ANAYA MULTIMEDIA

Desarrollo Web con PHP y MySQL

Luke Welling
Laura Thomson

LIBROS, MANUALES, CURSOS, REVISTAS, SOFTWARE, VIDEO TUTORIALES, TRUCOS, PELÍCULAS, SERIES, JUEGOS, UTILIDADES, Y MUCHO MÁS… GRATIS PARA EL CONOCIMIENTO CONTINUO.

https://www.facebook.com/IngAlexisLlontop

VISÍTANOS EN NUESTRAS DIFERENTES WEBS PARA DIFERENTES SERVICIOS!

LIBROS, MANUALES, SOFTWARE SOBRE PROGRAMACIÓN Y OTRAS UTILIDADES AQUÍ:

Compartiendo Conocimiento

http://alexyniorlls.blogspot.com

TODA INGENIERÍA EN UNA SOLA COMUNIDAD, LIBROS UNIVERSITARIOS Y DE TODA INGENIERÍA AQUÍ:

FULL ENGINEERING BOOK

http://fullengineeringbook.blogspot.com

REVISTAS, PROGRAMAS FULL AQUÍ:

ReProFull

http://reprofull.blogspot.com

PROGRAMACIÓN

Ttnn.o DE LA.QBIIAORIGINAL:
PHP and MySQL Web Developmeat

RP,otquLE Eonoáw.:
Víctor Manuel Ruiz Calderón
Susana ICrabe Pá'ez·R'!.!>in;

TUDUCCION:
José Luis Gómez Celador

ILUSTIACIÓNDE CUBIUTA:
Cecilia Poza Melero

EL_HOLISTICO@HOTMAIL.COM - 2006

Desarrollo web con PHP y J\1YSQL

Luke Welling
Laura Thomson

ANAYA MULTIMEDIA

Todal..... prapios..... m,, opendw&,equipaa ..... i!.
--~--lilInt&NIDila& d!J-lll&peClivM~DIf Me&
..... ;, ...

..... &le...-....
..... la..,., ..
..... ,,, ...
• 151 wwwa;adi Ir E f faa11 ,-
~,... i. 7 4-.--awpra01,J~
-. ....
Silmria.
:::-,,u ,,..J..... -•mur,,.
iIl'¡ ,, s,.....,. .. la,,...,_
v
*'
AUdlorincl crmlalionfmn Eqliab laDpqe
tdbioa; *flDlitletJPHP OIIIIJlySQL Wn Dnc/.,,,,.,.1,*
3111 Editioa by W.W.,. Lake; Tbomloa. Lura,
pablisbed by~ Edacetioa, lac, pubtiibing. Sama,
Copyripa O 200S by S... Publillmag.

~ .....
0- IDICIONES ANAYA MULTIMBDIA (GllUPO ANAY~ S.A.). 2085
-. lplcieuca deTeaa.15. 28027 Madrid
01,11·1, lepl:M.6.J20.ioo,
ISBN:Ml41S-llll-l
.....{....
.,=J .. lee«AIIIIGd&CII0..... S.L
::;~32.~)Wrid

A nuestros padres

EL_HOLISTICO@HOTMAIL.COM - 2006

Agradecimientos

Nos gustaría agradecer al equipo de la editorial su ttabajo. En concreto, nos gustaría darle las gracias a Shelley Johnston sin cuya dedicación y paciencia no se podría haber realizado este libro. Tambil!n a Israel Denis Jr. y Chris Newman por sus valiosas colaboraciones. El trabajo de los equipos de desarrollo de PHP y MySQL también ha sido magnifico. Nos ha facilitado la vida durante años y en la actualidad continúan haciéndolo.

Gracias tambiIn a Adrian Cose de eSec por decir en 1998"Eso se puede crear en PHP". Nos dijo que nos gustaría PHP y parece que tenía razón.

Por último, agradecer a nuestras familias y amigos por aguantar nuestra poca sociabilidad durante casi un año. En concreto, agradecemos el apoyo de nuestros familiares: Julie, Robert, Martín, Lesley, Paul, Adam, James, Archer y Qarton.

Acerca de los autores

Laura Thomson es profesora de la School of Computer Science and lnformation Technology en la RMIT University de Melboume, Australia. También es socia de la ga)ardonada empresa de desarrollo Web Tangled Web Design, Anteriormente ha trabajado para Telstra y el Boston-Cef151111iRg~BsJiceoci•da '°°'- hc>.nores en Ciencias Aplicadas (Ciencias Informáticas) y en Ingeniería (Ingeniería de sistemas informáticos). Actualmente completa su doctorado en Sitios Web adaptables. En su tiempo libre le gusta dormir. Su correo electrónico es lauraetangledwe. com. au.

L'uk~rWellinges profesor dif la Si:ñool of Computer Science and IiüoriilatióiC ; Technology en la RMIT Uni.versity de Melbourne, Australia. También es socio de Web Tangled Web Design. Es licenciado en Ciencias Aplicadas (Ciencias Informáticas) y actualmente completa un doctorado en Algoritmos generales para diseño de redes de comunicación. En su tiempo libre intenta perfeccionar su insomnio. Su correo electrónico es lukeetangledweb. com. au.

Ambos autores cuentan con el título MySQL Core Certification que ofrece MySQL Ab y con el título Zend Certified PHP Engineer, de Zend Technologies Ltd,

Acerca de los colaboradores

Israel Denis Jr. es un consultor autónomo que participa en proyectos de comerqo" electJónico por todo el mundo. Está especializado en la integración de paquetes SAP y Lawson, con soluciones Web personalizadas. Cuando no disefta ttware o_esaibe libros, le gusta viajar a Italia, dónde se siente como en casa. Desde 1998 cuenta con un Master en Ingeniería Electrónica que obtuvo en la Georgia Tech d• Atlanta, Georgia. Ha escrito numerosos artfculos sobre Linux, Apache,

PHP y MySQL. Ha trabajado con empresas como GE y Procter & Cambie, principalmente con sistemas informáticos basados en Unix. Su correo electrónico es ideniseureach .com.

Chris Newman es programador consultor especializado en el desarrollo de aplicaciones de Internet dinámicas. Dispone de una amplia experiencia comercial en el uso de PHP y MySQJ. para crear distintos tipos de aplicaciones para una base de clientes intemadonales. Licenciado por la Keele University, Chris vive en Stoke-on-Trent, Inglaterra, donde se encarga de Lightwood Consultancy Ltd., la empresa que fundó con su padre en 1999para ampliar su interés en el desarrollo por Internet. Chris quedó fascinado por el potencial de lntemét cuando estaba en la universiclad y su pasión es trabajar con la tecnología más puntera. Encontrará más info~dón sobre su empresa en la dirección http://www.lightwood.net. El correo electrónico de Chris Newman es chriselightwood.net.

EL_HOLISTICO@HOTMAIL.COM - 2006

Índice de contenidos

I ~u~ 39
R'uaNI pua leereste libro 31
Objetivol que puedealcanzar con eate libro 32
-aePHP 33
CoaceptodeMJSQL
pam.u.tilizu.lH,-~
Alpna de luen tidactes de PHP
Inlepdm de.,... de datos
Bibliomcaincorporada
Coete
Aprendizll¡e de PHP -
Compatibtlkt con el enfoque orientado objelloe
Portabilidad 36
C6digo fuente 36
Disp,nilrilidad de asistencia lmlice 36
Ncwedades dePHP 5.0
AJgaw de la ventaja de MJSQL-
Faciliacl de IIIO
Portabilidad
Coqigorunte
Dispaanlid.tde..... lbica
Noftdacles deMySQL SJJ

EL_HOLISTICO@HOTMAIL.COM - 2006

Organizad6n del libro 39
Conclusi6n 39

Parte l. UtilizarPHP 41

t. Cuno acelerado de PHP 42
Utilizar PHP 44
Crear una aplicación de ejemplo: Bob's Auto Parts 44
Crear el formulario de pedidos 45
Procesar el formulario 46
Inaustar PHP en HTML 47
Utilizar etiquetas PHP 48
Estilos de etiquetas PHP 48
Instrucciones de PHP 49
Espacios en blanco 50
Comentarios 50
Agregar contenido dinimico 51
Invocar funciones 52
Utilizar la función date() 52
Acceder a variables de formuJario 53
'latiables de fonmdario 53
Concatenar cadenas 56
'Variables y literales 57
Identificadores 57
variables-dedaradupor elusuario 58
Asignar valores a variables 58
npos de variables 58
Tiposde datos de PHP 59
Control de tipos 59
Convertir tipos 60
Variables de tipo variable 60
Declarar y utilizar constantes 61
Ámbito de variables 61
Utilizar operadores 63
Operadores aritmiticos 63
Operadores de cadena 64
Operadores de asignadón 64
Devolver valores de asignadón 64
Combinar operadores de asignación 65
Incremento y decremento previo y posterior 65
'Referendu 66
dores de comparadón 67
El operador iguales 67
Otros operadores de comparad6n 67
Operadores lógicos 68
Operadores bit a bit 69
Otros operadores 69
Operador ternario 69
Operador de supresión de error 70
Operador de ejecución 70
Operadores de matriz 70
El operador de tipo 71
Utilizar operadores: calcular los totales de los formularios 71
Precedencia y asociatividad: evaluar expresiones 73
Utilizar funciones de variables 74
Probar y establecer tipos de variables 74
Probar el estado de las variables 76
Reinterpretar variables 76
Implementar esl!Ucturas de control 77
Tomar decisiones con estructuru condicionales 77
Instrucciones if 77
Bloques de código 78
Instrucciones else 78
Instrucciones elseif 79
Instrucciones switch 80
Comparar condiciones diferentes 82
Iteración repetir .amones 82
85
Salir de una estructura de control o una secuencia de comandos 86
Utiliza una sintaxis alternativa de "estructuras de control 86
Utilizar declare 87
Siguiente puo: guardar el pedido del cliente 87

2. Almacenar y recuperar datos 88
Guardar datos para su lectura posterior 90
Almacenar y recuperar los pedidos de Bob 90
Procesar archivos 90
Abrir un archivo 91
Seleccionar modos de archivo 91
Utilizar topen0 para abrir un archivo 74
Abrir archivos a trav& de FTP o H'l1'P 91
Problemas al abrir el archivo 96
Escribir en un archivo 7
Parmnetros de fwrite() 8
Formatos de archivo 99
Cerrar un archivo 99

EL_HOLISTICO@HOTMAIL.COM - 2006

Leer desde un archivo 101
Abrir un archivo para su lectura: fopen() 102
Saber cuándo parar: feof() 103
Leer linea a linea: fgets(). fgetss() y fgeksv() 103
Leer todo el archivo: readfile(). fpusthru(). file() 104
Leer un cancter: fgek() 105
Leer una longitud arbitraria de bytes: fread() 106
Otras funciones de archivo útiles 106
Comprobar la existencia de un archivo: file_exists() 106
Averiguar el tamai\o de un archivo: filesize() 106
Eliminar un archivo: unlink() 106
Desplazarse dentro de un archivo: rewind(), fseek() y ftell() 107
Bloquear archivos 108
La opción más acertada: los sistemas 'le administración de base de datos 109
Problemas con el uso de archivos planos 109
Resolver estos problemas con RDBMS 110
Lecturas adicionales 111
A continuación 111

3. Utilizar matrices 112

¿Qué es una matriz? 114
Matrices indexadas numéricamente 114
Inicializar matrices indexadas numéricamente 115
Acceder a los contenidos de matrices 115
Utilizar bucles para acceder a la matriz 116
Matrices con diferentes indices 117
Inicializar una matriz 117
Acceder a elementos de matriz 117
Utilizar bucles 118
Operadores de matriz 119
Matrices multidimensionales 120
Ordenar matrices 124
Utilizar sort() 124
Utilizar asort() y ksort() para ordenar matrices 124
Invertir el orden 125
Ordenar matrices multidimensionales 125
Ordenaciones definidas por el usuario 125
Ordenaciones de usuario inversas 127
Reordenar matrices 128
Utilizar shuffle() 128
Utilizar arr~y_reverse() 129
Cargar matrices desde archivos 130
Otras manipulaciones de matrices 133
Desplazarse dentro de una matriz con each(). cunent(). reset(). end(). next(), pos() y prevO 133

Aplicar una función a cada elemento de una matriz: array_walk() 134
Contar elementos de una matriz: count(). sizeof() y array_count_values() 135
Convertir matrices en variables escalares: extract() 136
Lecturas adicionales 138
A continuación 138

4. Manipular cadeau y expresiones regulares 140

Ciear una aplicación de ejemplo: Smart Form Mail 141
Aplicar formato a cadenas 144
Umpiar cadenas: chop(). ltrim() y trim0 144
Aplicar formato a cadenas para presentaciones 144
Utilizar formato HTML: la función nl2br() 144
Aplicar formato a una cadena para su impresión 145
Cambiar mayúsculas y minúsculas en una cadena 147
Aplicar formato a cadenas para su almacenamiento: addslashes() y stripslashes() 148
Combinar y dividir cadenas con funciones de cadena 150
Utilizar-explode(). implode() y join() 150
Utilizar strtok() 151
Utilizar substr() 152
Comparar cadenas 153
Ordenarcadenas: strcmp(), strcasecmp() y strnatcmp0. 153
Comprobar la longitud de una cadena con strlen() 1M
Buscar subcadenas y reemplazar las con funciones de c~ 15'
Buscar cadenas en cadenas: strstrO. strchrO, strrchr() y stristr(). 155
Buscar.la posición de unasubcadena: strpos()...smpos(). 156
Sustituir subcadenas: str_replace() y substr_replace() 15
Introducción a las expresiones regulares 15
Los fundamentos 15
Conjuntos y clases de caracteres 59
ones 60
Recontar subexpresiones 161
Anclajes al principio o al final de una cadena 161
Bifurcación 161
Buscar coincidencias de caracteres especiales 162
Resumen de los caracteres especiales 162
Utilizar estos elementos en Smart Form 163
Buscar subcadenas con expresiones regulares 16'
Sustituir subcadenas con expresiones regulares 165
Dividir cadenas con expresiones regulares 165
Comparar ~ones de cadenas y funciones de expresiones
Lecturas adicionales
A continuación

EL_HOLISTICO@HOTMAIL.COM - 2006

S. Reutilizar c6di10 y crear funciones 168

Reutilizar código 170
Costes 170
Fiabilidad
Unifonnidad 170
Utilizar require() e include() 171
Utilizar require() 171
Extensiones de nombre de ardúvo y require() 172
Etiquetas de PHP y require() 172
Utilizar require() para plantillas de sitios Web 173
Utilizar include() 177
Utilizar require_cmce()e include_once() 178
Utilizar auto_prepend_file y auto_append_file 178
Utilizar las funciones de PHP 179
Llamar funciones. 179
Llamar a una función no definida 180
Funciones y el uso de mayúsculas y mimlsculas 181
Por qu6 debería crear funciones propias? 182
Estructura b4sica de una función 182
Designar funciones 183

!:~r~ padme~

Uamadas por referencia frente a llamadas por valor 189
Devolver desde funciones 190
Devolver valores ~esde funciones
Bloques de código
Implementar ~ión 193

1:::::n~~

6. PBP orientado a objetos-- 196
Conceptos orientados a objetos 198
Clases y objetos 198
Polimorfismo 199
Herencia 200
Crear clases, atributos y operaciones en PHP 200
Estructura de una clase 201
Constructóres 201
Destructores 202
c~ Instancias de clases 202
Utilizar atributos de clase 203
Controlar.el acceso con private y public- 205
°Llamaroperaciones de clase 206
Implementar la herencia en PHP 206

Controlar la visibilidad a trav& de la herencia por medio de private y protected '1111
Reeanplazos 209
Evitar la herencia y los reemplazos con final 210
Herencia múltiple 211
Implementar interfaces- 211
Diseñar clases 212
Escn"birel código para nuestra clase 213
Nuevas funciones avanzadas orientadas a objetos de PHP 5 221
PHPt frente a PHP 5 221
Utilizar constantes de clase 222
Implementar m6todos est4ticos 222
Compn>barel tipo de clase y sugerir tipos 223
Cl(mar objetos 223
Utilizar clases abstractas 22'
Sobrecargar m6todos con _call() 22f
Utilizarautoload() 225
lmpleinentar iteradores e iteraci6n 226
Convertir clases en cadenas 228
Utilizar el API de reflexión 228
A continuación- 229

7. Controlar excepciones 230
Conceptos del control de exc:epc:iones 231

~:=das por él U51JU'io,

Excepc:iones en el ejemplo Bob's Auto Parts
Excepciones y otros mecanismos de control en errores de PHP
Lecturas adicionales
A continuación'

Parte 11. Utilizar MySQL- 243

8. Dise6ar una base de datos Web 244
Conceptos de base de datos relacionales

~:~!:nas:

Filas- 2f7f.'J-
Valores-
Claves
Esqueaas 248-~
Relaciones- it?;~
Diseftar nuestra base de datos Web 2t~~~
Pensar en los objetos del mundo real que se estm modelando

EL_HOLISTICO@HOTMAIL.COM - 2006

Evitar el almacenamiento de datos redundantes 251
Utilizar valores de columna atómicos 252
Seleccionar claves lógicas 253
Reflexionar sobre las preguntas que desea formular a la base de datos 253
Evitar diseños con varios abibutos vados 253
Resumen de los tipos de tablas 254
Arquitectura de base de datos Web 254
Arquitectura 254
Lecturas adicionales 256
A continuación 256

9. Crear la base de datos Web 258

Utilizar el monitor de MySQL 261
Iniciar sesión e,;l MySQL 261
Crear bases de datos y usuarios 262
Crear la base de datos 263
Definir usuarios y privilegios 263
Introducción al sistema de privilegios de MySQL 263
Principio de asignación del privilegio más bajo 264
Configurar usuarios: el comando GRANT 264
Tipos y niveles de privilegio 266
El comando REVOKE 268
Ejemplos de uso de GRANT y REVOKE 268
Configurar un usuario para la Web 269
Cerrar la sesión como usuario rafz 270
Utilizar la b~~e 270
Crear tablas de base de datos 271
Significado del resto de las palabras dave- 272
Tipos de columna 273
Examinar la base de datos con SHOW y DESCRIBE 275
Crear índices 275
Nota sobre los tipos de tablas 276
Identificadores de MySQL 276
Seleccionar tipos de dato de columna 277
Tipos numéricos 278
Tipos de fecha y hora 279
Tipos de cadena 280
Lecturas adicionales 282
A continuación 282

10~_Trabajar con la base de datos de MySQL 284

Concepto de SQL 286
Ailadir datos a la base de datos- 286
Recuperar datos de la base de datos 288

Recuperar datos con criterios específicos 290
Recuperar datos desde varias tablas 292
Combinaciones sencillas de dos tablas 292
Combinar varias tablas 294
Buscar fil&:& que no coincidan 294
Utilizar otros nombres para designar tablas: los alias 296
Resumen de los tipos de combinación 296
Recuperar datos con un orden dado 297
Agrupar y agregar datos 298
Escoger las filas que recuperar 300
Utilizar subconsultas 300
Subconsultas búicas 301
Subconsultas y operadores 301
Subconsultas relacionadas 302
Subconsultas de filas 302
Utilizar una subconsulta como tabla temporal 303
Actualizar registros de la base de datos 303
Alterar tablas tras su creación 304
Eliminar registros de la base de datos 306
Eliminar tablas 307
Eliminar una base de datos entera 307
Lecturas adicionales 307
A continuación 307

11. Acceder a la base de datos de MySQL desde la Web coaJtBP 308

Consultar una base de datos desde la Web 3
Comprobar y filtrar datos entrantes 3
Configurar una conexión 31
Seleccionar una base de datos 31~
Consultar la base de datos 31!C
Recuperar resultados de consulta 31~~
Desconectarse de una base de datos 319
Añadir nueva información a la base de datos 319
Utilizar instrucciones predefinidas 323
Utilizar otras interfaces de bue de datos y PHP 324
Utilizar una ~terfaz de base de datos genérica: P~ 08 ~24-§
Lecturas ad1aonales 327
A continuación 31

12. Administraci6n avanzada de MySQL- 32B

Anilisis detallado del sistema de privilegios 329
La tabla user 330

EL_HOLISTICO@HOTMAIL.COM - 2006

Las tablas db y host _: .. 332

Las tablas tables_priv y colUD111S_priv ..--334
Control de acceso: cómo utiliza MySQL las tablas de concesión
de privilegios 335
Actualizar privilegios: cuúulo surten efecto los cambios .. 335
Proteger la base de datos MySQL 336
MySQL desde el punto de vista del sistema operativ0 -336
Contraseñas 336
Privilegios de usuario 337
Problemas relacionados con la Web 338
Obtener mú informadón sobre bases de datos 338
Obtener infonnaci6n con SHOW 339
Obtener informadón sobre columnas con DESCRIBE 340
Fundonamiento de las consultas con EXPLAJN 340
Agilizar consultas con Indices 346
Optimizar una base de datos 346
Optimizar el disea\o 346
Permisos 347
Optimizar tablas 347
Utilizar índices 341
Utilizar valores predeterminados 347
Otras sugerencias 347
~!Y.@r ~a. copia de seguridad de la base de datos MySQL 348
RestabJecer la base de datos MySQL 348
Im1tle111entar la replicadón 349
Configurar el servidor prindpal 350
Realizar la transferencia de datos inidal 350
Configurar el servidor o servidores secundarios 351
Lecturas adidonales 351
A continuad6n 352

13. Programaci6n avanzada con MySQL 354

La instrucd6n LOADDATA INFILE 355
Motores de almacenamiento 356
Transacciones.- 357
Definir transacdones 357
Utilizar transacdones con lnnoDB 358
Claves secundarias 359
Procedimientos almacenados 360
EjeJnplo básico 361
Variables locales 363
C~ y estructuras de control 364
Lectma ádidonales 367
A continuad6n J67

Parte 111. Comercio electr6nico y seguridad 369

14. Crear un sido Web de comercio electr61lico 370

Decidir un objetivo 371
npos de sitios Web co111erciales 312
Medios publidtarios en línea 372
No suministrar información importante 373
Mala presentadón 373
No responder a la información generada por el sitio Web 373
No actualizar el sitio 374
No realizar el seguimiento del éxito del sitio 374
Recoger pedidos de artículos y servidos 375
Preguntu sin respuesta 377
Confianza 377
Facilidad de-uso 378
Compatibilidad 379
Suministrar servicios y artículos digitales ~79
Añadir valor a los artículos y servidos 380
Recortar costes 380
Riesgos y amenazas 381"
Piratas infonn6ticos 382
Fracaso en la atracción de suficiente negocio 382
Fallos de hardware ínformático 383
Fallos de alimentación, comunicación. redes y distñ"tiüa6ñ a: 1
Cmnpetenc:ia excesiva 383
Errores de software
Cambios en las políticas e impuestos gubernamentales
Umites de la capacidad del sistema 384
Seleccionar estrategia 385
A continuad6n

15. Aspectos de seguridad reladonados con el C01Dercio electr6nlco 38(

Importanda de la informadón 388
Amenazas contra la seguridad 388
Exposici6a de datos confidendales 389
Pérdida o destrucción de datos 391
Modificar datos 391
Negad6ndeservido 392
Errores en el software 393
Malas especificadones tfalicas 393
Supndrinnes erróneas héc:has por los desanolladom 394
Pruebas incmnpletas 3
Repudio
Equilibrio entre usabilidad, rendimiento, coste y segundad 75

EL_HOLISTICO@HOTMAIL.COM - 2006

Crear una politica de seguridad 396
Principios de autenticación 397
Utilizar autenticación 398
Fundamentos de la encriptación 398
Encriptación de clave privada 400
Encriptadón de clave pública 400
Firmas digitales 401
Certificados digitales 402
Servidores Web seguros 403
Auditorias y registros 404
Cortafuegos 405
Copia de seguridad de los datos 405
Copia de seguridad de archivos generales 406
Copia de seguridad y restauración de bases de datos MySQL 406

16. Implementar autenticación con PHP y MySQL 408

Identificar visitantes 409
Implementar el control de accéso 411
Almacenar contrasei\as 413
Cifrar contraseftas 416
Proteger p6ginas múltiples 417
Autenticación báica 418
Utilizar autenticación báica en PHP 419
Util~ar autenticación báica con los archivos .htaccess dt Apache 421
Autenticación b6sica con IIS 424
Utilizar autenticación mod_auth_mysql 427
Instalar mod_auth_mysql 427
¿Funcionó? 428
Utilizar mod_auth_mysql 428
Crear un sistema de autenticación propio 429
Lecturas adicionales 429
A continuación 430

17. Implementar transacciones seguras con PHP y MySQL 432

Suministrar transacciones seguras 433
El equipo del usuario 434
Internet 436
Su sistema 437
Utilizar SSL 438
FUtrar lu entradas de los usuarios 441
Proporcionar un almacenamiento seguro 442
Determinar si almacenar o no números de tarjeta de crédito 443

Utilizar encriptación en PHP 444
Instalar GPG 445
Probar GPG 447
Lecturas adicionales 452
A continuación 452

Parte IV. Técnicas avanzadas de PHP 453

18. Intenctuar con el sistema de archivos y el servidor 454

Introducción a la carga de archivos 456
HTML para la carga de archivos 456
Un inciso sobre seguridad 457
Código PHP para procesar la tarea de carga del archivo 457
Problemu habituales 461
Utilizar lu funciones de directorio 462
Leer desde directorios 462
Obtener información sobre el directorio actual 464
Crear y eliminar directorios. 464
Interactuar con el sistema de archivos 465
Obtener información sobre archivos 465
Cambiar propiedades de archivo 468
Crear, elinúnar y desplazar archivos 468
Utilizar funciones de ejecución de programas 469
Interactuar con el entorno: getenv() y putenv() 472
Lecturas adicionales 472

Descripción general de los protocolos disponibles 4~
En~r y recib~ ~orreos electrónicos 476
Utilizar otros sitios Web 477
Utilizar funciones de búsqueda de red 479
Utilizar FTP 483
Utilizar FTP para realizar una copia o reproducir un archivo 484
Conectarse al servidor FTP remoto 487
Iniciar sesión en el servidor FTP 487
Comprobar el tiempo de actualización de archivos 481
Descargar el archivo 489
Cerrar la conexión 490
Cargar archivos 490
Ev~t~r tiempos de ~pera 490
Utilizar otras funciones de FTP 491
Lecturas adicionales 4
A continuación

EL_HOLISTICO@HOTMAIL.COM - 2006

20. Administrarla fechay la bora — — —--, ... 494

Obtener la fecha y la hora en PHP ····--- -.''..--·····-·····--·-·-·-····-- 495
Utilizar la función date() ·-·-··--····· ·-··-·-·-·····-·-·-··-····· ..—..495
Marcu de tiempo de Unix -..-..--·····--····-·-·---- · ·-·-·-·- - ..497
Utilizar la función getdate() -·--····---- ..·..···--·-·—·-·--·-···· ..·-···—··- 499
Validar fechas---·····..·····-·-····-·-·-·····-····· --·····..···-···-500
Convertir entre formatos de fecha de PHP y MySQL ·-- - ——— - ..... - ..500
Calcular fechas en PHP —·—·-·····-·-·-···--- ..-- -·-·-·—·-··· 502
Calcular fechas en MySQL.•·-·-··--·····-·-·--····· ..·-·-···..—·-··-..·-·-···- 503
Utilizar microsegundos ·-····--..·····-····-····-··· -·—·--·· • ·--·-·- 505
Utilizar funciones de calendario ·-·····-···..-·····-····· ..·505
Lecturu adicionales ·-·-·····- ..··--·····——·· ..'''''-·~---·'·—'''---·--·--'''' 506
A continuación ...·——·-···- ..··-·····-···—·-·· ..-·-·-·-·---— — —-·-·-···-·- 506

21. Generar imigenes — — — ·-· —— - — — — — — — — — — -- 508

Configurar la compatibilidad de i!Mgenes en PHP : 510
Formatos de imagen ·-·····-··-·····-···-····· 511

J.~==='=,=NNNi ,=:'=-m

Crear un lienzo ·—-····· , 514
Dibujar o imprimir texto en la imagen - 514
~===r:!:! cofinah __:, .---, ''''= ''''~-~~~

•Utilizar imágenes generadu autmnAticamente en otru páginas ..:::517
Uti~zar ~to y fuentes para crear im6genes .•..., ·-·-·—·····---···-···--518
A'~ el li~ base 522
,ustar texto en e botón ·····-·-····· - --522
Colocar el texto ---····-····· ..··-·····——····· 525
~~bir el texto en el botón _ _ __::526
Dib~ra :rminar·--·-·····-····· ·-·'-····-·-·- ..—·-·--526
Otr ular guru y representación gr6fica de datos ..·-·--·-·-·-·· ..·--·····-·-·----527
i.ec:~~~~imagen ·--·-·--·--·--·--·-·---···534

~~·:~ contin~~ --..~-:-·····-·· ..·.. ···---·-----····-·-·-·-·

DocuJnentar nuestros proyectos 577
Prototipos 577
Sep~ra~lógica l' ~ontenido 578
Optimizarel código 579
Utilizar una optimización sencilla 579
Utilizar productos Zend 580
Pruebas 580
Lec~ adi~ionales 582
A continuaaón 582

25. **Depuración** 584

Errores de programación 585
Errores sin~ctico_s 586
Errores de eiecuaon 587
Llamadas a funciones que no existen 588
Lectura o escritura de archivos 589
Interacción con MySQL u otras bases de datos 590
Conexiones a redes de servicios 591
Comprobar incorrectamente los datos de entrada 592
Errores lógicos 592
Ayuda de depuración de variables 594
Niveles de informes de error.es 596
Modificar los parámetros de informesde error 597
Desencadenar errores propios 599
Solucionar errores con elegancia 599
A continuación 60z

26. Generar autenticación y personalización de usuarios 604

El problema 606
Componentes de la solución 606
Identificar y personalizarusuarios 606
Almacenar marcadores 607
Recomendar marcadores 607
Presentación de la solución 608
Implementarla base de datos 610
Implementarel sitio básico 611
Implementarla autenticación de usuarios 614
~~tro 614
Iniciar Se51ón 620
Cerrarsesió~ 623
Modificar contraseñas 624
Restablecer contraseñas olvidadas 626
Implementarel almacenamiento y la recuperación de marcadores 631
Aftadir marcadores 631

Mostrar marcadores 633
Eliminar marcadores 634
Implementar recomendaciones 636
Conclusión y posibles ampliaciones 640
A continuación 640

27. Crear un carro de la compra 642

El problema 644
Componentes de la solución 644
Generar un catálogo en lfnea 644
Realizar el seguimiento de las compras de un usuario mientras compra 644
Implementar un sistema de pago 645
Diseiiar una interfaz de administración 645
Presentación de la solución 646
Implementar la base de datos 650
Implementar el catálogo en lfnea 652
Enumerar categorías 653
Enumerar los libros de una categoría 656
Mostrar detalles sobre un libro 658
Implementar el carro de la compra 659
Utilizar la secuencia de comandos show_cart.php 659
Ver el carro 663
Añadir artículos al carro 665
Guardarel carro actualizado 667
Imprimir un resumen en la barra de encabezado 667
Salir 668
Implementar el pago 673
Implementar una interfaz de administración 6!?.
Ampliar el proyecto 683
Utilizar un sistema existente 684
A continuación 684

28. Crear un sistema de administración de contenidos 686

El problema 687
Requisitos de la solución 688
Sistemas existentes 688
Modificar el contenido 689
Añadir contenido en el sistema 689
FTP/SCP 689
Método de carga de archivos 689
Cambios en línea 689
Bases de datos frente a almacenamiento en archivos 690
Estructura de documentos 690
Utilizar metadatos 69(f

EL_HOLISTICO@HOTMAIL.COM - 2006

Aplicar formato al resultado.--.-- 691
Oileftar y presentar la solución 692
Oileftar la base de datos- 693
Implementar el sistema 695
Interfaz de usuario. 695
Manipular imigenes 699
Sistema 702
Búsquedas 710
Pantalla del edilor 714
Ampliar el proyecto 715
A continuación 716

29. Crear ua servicio de correo electrónico basado ea la Web -- 718

El probleala 719
Componentes de la solución 720
Presentación de la solución 722
Configurar la base de datos 723
Arquitectura de la secuencia de comandos 725
Iniciar y cerrar sesión 730
Configurar cuentas 734
Crear una nueva cuenta 736
Mtidlf.car mia cuenta aiatente 737
Eliminar una cuenta 737
Leer correo 738
Seleccionar una cuenta 738
Ver los cé>ñtenidos-delbuzón de- torreo 741
Leer un mensaje de correo 744
Ver encabezados de mensaje 747
Eliminar correo 748
Enviar correo 749
Enviar un nuevo mensaje 749
Responder o reenviar correo 751
Ampliar el proyecto 753
A continuación 754

30. Crear ua gestor de listas de corre• 756

El problema 758
Compcnentes de la solución 758
Definir una base de datos de listas y suscriptores 758
Cargar archivos 759
~viar correo con archivos adjuntos 759
Presentar la solución 760
Definir la base de datos 762
Arquitectura de la secuencia de comandos 764

Implementar el inicio de sesión 772
Crear una nueva cuenta 772
Iniciar sesión 775
Implementar funciones de usuario 778
Ver las listas 778
Ver información de listas 782
Ver archivos de listas 785
Realizar y anular suscripciones 786
Modificar la configuración de una cuenta 787
Modificar contaseñas 787
Cerrar sesión 789
Impleaaentar funcioaes administrativas 790
Crear una nueva lista 790
Cargar un nuevo boletín informativo 792
Procesar la carga de varios archivos 795
Vista previa del boletín informativo 799
Enviar el mensaje 801
Ampliar el proyecto 806
A continuación 807

31. Cnar foros Web --- 808

El problema 809
Componentes deJa sohic:ión 810
Presentar la solución 812
Disei\ar la base de datos 813
Verel árbol d~artículos
Desplegar y replegar
Mostrar los artfaalos
Utilizar la clase treenode
Ver artículos individuales 828
Atladir nuevos artfculos 830
Añadir ampliaciones 836
Utilizar un sisteJna existente 837
A continuación 837

32. Generar documentos penonallzados ea formato PDF 838

El pioblema 839
Evaluar formatos de documento 840
Papel- 840
ASCII 841
HTML 841
Formatos de procesadores de texto 841
Formato de texto enriquecido 842
PostScript 841

EL_HOLISTICO@HOTMAIL.COM - 2006

Formato de documento portable 843
Componentes de la solución 844
Sistema de preguntas y respuestas 844
Software de generación de documentos 844
Software para crear una plantilla RTF 845
Software para crear una plantilla PDF 845
Software para crear PDF mediante programación 846
Presentar la solución 847
Responder a las preguntas 848
Calificar las respuestas 849
Generar un certificado RTF 852
Generar un certificado PDF a partir de una plantilla 856
Generar un documento PDF por medio de PDFlib 859
Una secuencia de comandos Helio World para PDFlib 859
Generar un certificado con PDFlib 864
Solucionar problemas con los encabezados 871
Ampliar el proyecto 871
Lecturas adicionales 872
A continuación 872

33. Conectane a servicios Web con XML y SOAP 874

El problema 875
Entender XML 876
Servicios Web 880
SOAP 880
WSDL 881
Componentes de la solución 882
Crear un carro de la compra 882
Utilizar las interfaces de servicios Web de Amazon 882
Analizar XML 883
Utilizar SOAP con PHP 883
Almacenar en caché 883
Presentación de la solución 8&1
Aplicación principal 888
Mostrar los libros de una categoría 894
Obtener un clase AmazonResultSet 896
Utilizar REST/XML sobre HITP 902
Utilizar SOAP 907
Almacenar los datos en cac:hi 908
Crear el carro de la compra 910
Conectar a Amazon 913
Instalú el código del proyecto 914
Ampliar el proyecto 915
Lecturas adicij>IUlles 915

Parte VI. Apádlces 917

Apéndice A. Instalar PHP y MySQL 918

Ejecutar PHP como in~rpreteCGI o coino módulo 920
Instalar Apache, PHP y MySQL en Unix 920
Instalación binaria 920
Instalación desde código fuente 921
Instalar MySQL 922
Instalar PDFlib 924
Instar PHP 924
Fragmentos de código de httpd.conf 928
¿Funciona la compatibilidad con PHP? 928
¿Funciona SSL? 930
Ultimo&pasos 930
Instalar Apache, PHP y MySQL en Windows 930
Instalar MySQL en Windows 931
Configurar la rut~ 933
Eliminar el usuano anónlDIO 933
Definir la contrasefta raíz 933
Instalar Apache en Windows 934
Instalar PHP en Windows 936
Aftadir PHP a su configuración de Apache 937
Aftadir PHP y MySQL a Mkrosoft OS y PWS 937
Pruebas 938
Instalar PEAR 939
Configuraciones-.dicionale. 940

Apéndice B. Recursos Web 942
PHP 943
Recunos específicos de MySQL y SQL 946
Recursos para Apache
Desarrollo Web 946

Apé•dice C. Contenido del CD-ROM 948
Windows 949
Unux/Unix 950

Índice alfaWtlco 951

EL_HOLISTICO@HOTMAIL.COM - 2006

Introduc<;-ión

Bienvenidos a PHP y MySQLDesarrollo Web. Estas p6ginas recogen los conocimientos fruto de nuestras experiencias con PHP y MySQl.;, dee de Ju herramientu de desarrollo Web mú actuales.

En esta introducción, trataremos los siguientes aspectos:

- Razones para leer este libro
- Objetivos que se pueden alcan7.arutilizando este libro
- Qué es PHP y MySQL,y por qué son unas herramientas geniales
- Una repaso general a las nuevas características de PHP 5 y MySQL5
- Organización del libro

Empecemos.

Razones para leer este libro

En este libro se explica cómo crear sitios Web interactivos, desde los formularios de pedidos más sencillos hasta los sitios de comercio electrónico mú ~piejos y seguros, y lo que es mú, con ayuda de temologfas de código abierto

Este libro va dirigido a lectores con un conocimiento bmco de HM'll. y con cierta experiencia de programación con len—jes actuales, aunque no es necesario

EL_HOLISTICO@HOTMAIL.COM - 2006

que hayan programado para Internet o utilizado una base de datos reladonal. Los lectores con menos experiencia en programación tambiin podr4n aprovechar este libro aunque tardar4n un ~ mú en digerir sus conceptos. Hemos intentado tratar los conceptos básicos, aunque se aborden rápidamente. Este libro va dirigido a personas que deseen dominar PHP *y* MySQL para crear un sitio Web comercial o de gran tamai\o. Puede que tenga experiencia con otros lenguajes de desarrollo Web. Si así fuera, este libro le ayudará a ponerse al día rápidamente.

La razón que nos ha animado a escribir este libro es que todos los manuales sobre PHP que conocíamos eran búicamente guías de referencia de func~nes. Estos libros resultan útiles pero no sirven para mucho cuando nuestro jefe o un cliente nos piden crear un carro de la compra para un sitio Web. Hemos intentado que los ejemplos resulten útiles. De hecho, gran parte de su código se puede utilizar directamente en un sitio Web o bastaría con pequeñas modificaciones para hacerlo.

Objetivos que puede alcanzar con este libro

La lectura de este libro le permitirá crear sitios Web reales *y* din6micos. Si ha desarrollado sitios Web utilizando simple HTML, ya conocer4 las limitaciones de este enfoque. El contenido estático de los sitios desarrollados únicamente con Hl'ML es exactamente eso, estático. Estos sitios no varían a menos que se actualicen físicamente. Los usuarios no pueden interactuar con el sitio de forma significativa.

El uso de un lenguaje como PHP *y* una base de datos como MySQL permite crear sitios Web dinámicos, es decir, susceptibles de personalización *y* dotados de información en tiempo real.

En este libro nos hemos centrado deliberadamente en aplicaciones del mundo real, incluso en los capítulos iniciales. Comenzamos por analizar un sendllo sistema de pedidos en linea y aprovecharemos para examinar los distintos.componentes de PHP y MySQL

Abordaremos aspectos relacionados con el comercio electrónico *y* la seguridad mientras desarrollamos un sitio Web real y le mostraremos cómo implementarlos en PHP y MySQL

En la sección final del libro, comentaremos cómo abordar proyectos del mundo real y le guiaremos a travs del diseño, planüicación y construcción de los siguientes ocho proyectos:

- Autenticar y personalizar usuarios
- Carros de la compra
- Sistemas de administración de contenido
- Correo electrónico basado en la Web
- AdDiinistradores de listas de correo
- PorosWeb

- Generación de documentos PDF
- Servicios Web con XML *y* SOAP

Puede utilizar estos proyectos directamente o modificarlos para ajustarlos a sus necesidades. Se han seleccionado porque creemos que representan las ocho aplicaciones Web mú comunes desarrolladas por programadores. Si sus necesidades son distintas, este libro deberla servirle de ayuda para lograr sus objetivos.

Concepto de PHP

PHP es un lenguaje de secuencia de comandos de servidor diseti.ado espedfic:amente para la Web. Dentro de una página Web puede incrustar código PHP que se ejecutará cada vez que se visite una página. El código PHP es interpretado en el servidor Web *y* genera código HTML *y* otro contenido que el visitante verá.

PHP fue concebido en 1994 *y* es fruto del trabajo de un hombre, Rasmus Lerdorf. Ha sido adoptado por otras personas de talento *y* ha experimentado cuatro importantes trasformacioneshasta convertirse en el producto actual. En agosto de 2004, se encontraba instalado en mú de 17 millones de dominios de todo el mundo *y* su número c:iec:e rápidamente. Si desea conocer el número actual de sitios que utilizan este lenguaje, visite el sitio ht tp: / /vww. php. net/uaage. php.

PHP es un producto de código abierto, lo que quiere decir que puede acceder a su código. Puede utilizarlo, modificarlo *y* redistribuirlo sin coste al—.

Las siglas PHP equivalían inicialmente a P—Jic,m~!'age.(~ágina !f! J!':.i~0. personal) pero se inóclificaron de acuerdo con la convención de designación de GNU (del ingls, Gnu·s Not Unix, Gnu no es Unix) *y* ahora equivale a PHP Hipe~t Preproc:essor(Preprocesador de hipertexto PHP). En la actualidad, PHP est6 .Jl' su versión 5. Esta versión incorpora mejoras importantes en lo que respecta al motor Zend subyacente *y* al lenguaje que utiliza. La dirección Web de la página de PHP está disponible es ht tp: // www. php. net. La dirección de la página de Zend Technologies se encuentra en http://www. zend. com.

Concepto de MySQL

MySQL es un sistema para la administración de bases de datos relacionales (RDBMS) rápido y sólido. Las bases de datos permiten almacenar, buscar, ordenar *y* recuperar datos de forma eficiente. El servidor de MySQL controla el acceso a los datos para garantizar el uso simultáneo de varios usuarios, para proporcionar acceso a dichos datos *y* para asegurarse de que sólo obtienen acceso a ellos los usuarios con autorización. Por lo tanto, MySQL es un servidor multiusuario *y* de subpror:esamiento múltiple. Utiliza SQL (del ingl& Structured Query LanguJge,

EL_HOLISTICO@HOTMAIL.COM - 2006

Lenguaje de consulta estructurado), el lenguaje estalndar para la consulta de bases de datos utilizado en todo el mundo. MySQL lleva disponible desde 1996 pero su nacimiento se remonta a 1979. Ha obtenido el galardón Choice Award del Linux Joumal Readers en varias ocasiones.

MySQLse distribuye bajo un sistema de licencias dual. Puede utilizarlo bajo una licencia-de código abierto (GPL), que es gratuita mientras cumpla las condiciones de la misma. Si desea distribuir una aplicación que no sea GPL y que incluya MySQL, puede adquirir una licencia comercial.

Razones para utilizar PHP y MySQL

Al desarrollar un sitio de comercio electrónico, se pueden utilizar una gran cantidad de productos düerentes:

- Hardware para el servidor Web
- Un sistema operativo
- Software de servidor Web
- Un sistema de administraciónde base de datos
- Un lenguaje de secuencia de comandos o de programación

Algunas de estas opciones dependen de otras. Por ejemplo, no todos los sistemas operativos se ejecutan sobre todo el hardware ni todos los lenguajes de secuencia de comandos se pueden conectar a bases de datos, etc.
--Eruste. libro-no...vamosa-prestar-mucha atenc:iórnd hardware~ni a tos sistemas operativos ni al software de servidor Web. No ser6 necesario. Una de las ventajas de PHP es que est6 disponible para Miaosoft Windows, para muchas veniones de Unix y para cualquiet servidor Web completamente funcional MySQLresulta ig1181-mente verútil. Para demostrarlo, los ejemplos de este libro se han escrito y probado en dos configuraciones muy utilizadas:

- Linux con el servidor Web Apache
- Microsoft Windóws XP con Miaosoft Internet lnformation Server (DS)

Sea cual sea el hardware, sistema operativo y servidor Web que elija, le recomendamos que considere seriamente la opción de utilizar PHP y MySQL

Alguna de las cualidades de PHP

Entre lÓs competidores principales de PHP se puede citar a Peri, Miaosoft Active Server Pages (ASP), Java Server Pages OSP) y Allaire ColdFusion.

En comparación con estos productos, PHP cuenta coa muchas ventajas, entre las que se encuentran las siguientes:

- Alto rendimiento
- Interfaces para düerentes de sistemas de base de datos
- Bibliotecas incorporadas para muchas tareas Web ~bituales
- Bajo coste
- Facilidad de aprendizaje y uso
- Portabilidad
- Disponibilidad de código abierto
- Disponibilidad de asistencia técnica

A continuación se comentan en m6s detalle estas cualidadés.

Rendimiento

PHP es muy eficaz. Mediante el uso de un úni~ servidor, puede serv~ millones de acceso al dfa. Los indicadores comparativos de rendimiento publicados ~ Zend Technologies (ht tp: / /www. zend. com) muestran que PHP supera ampliamente a sus competidores en esta faceta.

Integración de b_~$e d~ <tatgs

PHP ... de una --- —— . ..... los ..... de .... ial

Adem6s de MySQL, puede conectarse directamente a las bases de datos PostgreSQL.mSQL, Orade, dbm, FilePro, Hyperwave, Inf~, Inter~ase y S~. entre otras. PHPS tambiin cuenta con una interfaz SQL incorporada a un archivo-plano denominada SQUte.

El~ de ODBC (del inglés Open Oatabase CoMectivity Standa~, Est6ndar ~e conectividad abierta de base de datos) permite establecer una conex1ó~a cualqwer base de datos que suministre un controlador ODBC. Entre ellas, se mduyen los productos de Microsoft, y muchos otros.

Bibliotecas incorporadas

Como se ha diseftado para su uso en~ Web, PHP in~orpora una gran cantidad de funciones inlegradas para realizar útiles tareas ":laaonadas con la '!eb· Puede enerar im6genes GIF al instante, establecer~ones a otros ~ de red~ .nviar correos electr6nicos, trabajar con cookies y generar d~ POF, tod~ COll'una& pocas lineas de código.

EL_HOLISTICO@HOTMAIL.COM - 2006

Coste'

PHP es gratuito. Puede descargar la última versión de ht tp: / /www.php. net cuando lo desee sin coste alguno.

Aprendizaje de PHP

La sintaxis de PHP se basa en otros lenguajes de programación, principalmente en C y Peri. Si ya conoce C o Peri, o un lenguaje de tipo C como C++ o Java, no tardar4 nada en utilizar PHP de manera productiva.

Compatibilidad con el enfoque orientado a objetos

La versión 5 de PHP cuenta con completas funciones orientadas a objetos. Si ha aprendido a programar en Java o C++ encontrarú las funciones que espera (y habitualmente la misma sintaxis), como por ejemplo herencia, atributos y métodos privados y protegidos, clases y métodos abstractos, interfaces, constructores y destructores. Incluso encontrará características menos comunes como un comportamiento de iteraciones incorporado. Algunas de estas funciones estaban disponibles en las versiones 3 y 4 de PHP, pero en la 5 la compatibilidad con el enfoque orientadp a objetos es mucho más completa.

Portabilidad

PHP es.i. disponible para una gran cantidad de sistemas operativml difei:entes. Puede escribir código PHP en todos los sistemas operativos gratuitos del tipo Unix, como Unux y FreeBSD, versiones comerciales de Unix, como Solaris e IRIXo en las diferentes versiones de Microsoft Windows. Su código funcionará sin necesidad de aplicar ninguna modificación a los diferentes sistemas que ejecute PHP.

Código fuente

Dispone de acceso al código fuente de PHP. A diferencia de los productos comerdales y de código cerrado, si desea modificar algo o agregar un elemento al programa, puede hacerlo con total libertad.

No necesitara esperar a que el fabricante publique parches, ni tendrá que preocuparse porque el fabricante cierre sus puertas o dedda abandonar el producto.

Disponibilidad de asistencia técnica

Zend Teéhnologies (www. zend. com), la empresa responsabledel motor de PHP, basa su desarrollo en la oferta de asistenda téauca y software de forma regular.

Novedades de PHP 5.0

Puedeque hayacambiado a PHP 5.0 de una de las versiones de PHP 4.0. Como es de esperar en cualquier versión importante, se han produado cambios significativos. El motor Zend de PHP se ha reelaborado por completo. Entre algunas de las nuevas características destacamos las siguientes.

- Una mejor compatibilidad con el enfoque orientado a objetos basada en un modelo de objetos completamente nuevo.
- Excepciones para la resoluoón de errores de forma mantenible y escalable.
- SimpleXML para procesar datos XML de forma sencilla

Entre otros cambios mencionamos el cambio de algunas de las ex~~nes predeterminadas de PHP a la bibliotl!ca PECL, la mejora de la compatibilidad con flujos y la inclusión de SQLite.

Algunas de las ventajas de MySQL

Entre los competidores principales de MySQL, se puede ~tar a PostgreSQL, Microsoft SQL~er yOracle. MySQL cuenta con muchas ventalas, e~tre las que se encuentran las descritas a continuación.

Rendimiento

MySQ~ es muy ~ ~ollaores ;n el sitio Web ht tp: //web. mysql . c~t comparativos de sus esa ch d'ferencia de velocidad abismal con:, Estos indicadores revelan en mu os c~ una respecto a los productos de la competenoa.

Bajo coste

MySQLes~ diseqbo~dibleendefo= ==~rcial. Necesitará una licencia si o por un preoo r. ua o desea redistribuir MySQL como parte de una aplicaaón y ~ q~~e em. leer una licencia de código abierto en la misma. Si no tiene pensa~o ~wr ~ p o traba)B en software gratuito no es necesario que adqwera una licenaa.

Facilidad de uso

Las bases de datos más modemu utilizan ~L. Si ha utilizado o"'::!'!:::: debería tener proble~s para aclaptarse a es~ 5:15tema. MySQLresul sencillo de configurar que otros productos similares.

EL_HOLISTICO@HOTMAIL.COM - 2006

Portabilidad

MySQL se puede utilizar en una gran cantidad de sistemas Unix diferentes así como bajo Microsoft Windows.

Código fuente

Como en el caso de PHP, puede obtener y modificar el código fuente de MySQL En la mayoría de los casos no es un aspecto importante para los usuarios pero es aco——ble ya que garantiza ~ continuidad en el futuro y le p~oporciona diferentes opaones en caso de emergencia.

Disponibilidad de asistencia técnica

No todos ~ pr~uct~ de código abierto cuentan con una empresa principal que ofrezca UJStenaa tbica, asesoramiento y certificación, aunque puede obtener todas estas ventajas de MySQL AB (www. mysql. com).

Novedades de MySQL 5.0

En MySQL 5.0 podemos apreciar importantes cambios, entre los que destacamos:

- Los procedimientos almacenados
- La compatibilidad con cursores

Entre otras de las novedades podemos mencionar la compatibilidad con el estAi:"1~ANSI_1. las mejoras de velocida4.

Sl sigue utthzan una de las primeras versiones 4.x del servidor MySQL o una versión 3.x, debería saber que desde la 4.0 han aparecido las siguientes novedades:

- Compatibilidad con subconsultas
- Tipos GIS para almacenar datos geográficos
- Compatibilidad me;orada con internacionalización
- El método de almacenamiento InnoDB compatible con transacciones se incluye como esUndar
- La cachetde consultas MySQL, que mejora considerablemente la velocidad de las consultas repetitivas que suelen ejecutar las aplicaciones Web.

Organización del libro

Este libro se divide en seis partes principales:

- Parte L Utilizar PHP: Proporciona una visión general de las partes fundamentales del lenguaje PHP. Los ejemplos escogid°.' son ejem,los del m~do real utilizados para desarrollar un sitio de comerao electrónico y no código sin utilidad práctica. Esta sección se inicia con el capitulo ·curso ª°:leraclo·. Si ya tiene experiencia en programación con PHP, puede repasar rápldamente esta sección. Si ha trabajado antes con PHP o acaba de llegar al mundo d~ la programación, conviene que se detenga un poco más sobre ella. Incl~ sl ya está familiarizado con PHP, deberla consultar el capítulo sobre PHP onentado a objetos, ya que en PHI'.estas funciones han.cambiado considerablemente.
- Parte D. Utilizar MySQL: Aborda los conceptos y el dlsei\o implicado en el uso de sistemas de bases de datos relacionales como MySQL, el uso de SQL, la conexión de la base de datos MySQLal mundo con PHP y temas avanzados sobre MySQL, como la seguridad y la optimización.
- Parte DI. Comercio electr6nico y aeguridad: Trata alguno de los te~as generales implicados en el desarrollo de un ~tio de comercio elec~co utilizando cualquier lenguaje. El aspecto más importante es la segun~ad. Seguidamente, explicaremos cómo utilizar PHP y MySQL para autenticar usuarios y recoger, transmitir y almacenar datos de manera segura.
- Parte IV. Tbical avanzadas de PHP: Trata de manera detallada alguna de las funciones principales que integra PHP. Se han seleccionado aquellos grupos de funciones que tíen~ una mayor JnOb:-bilidad de-aparición--:1'~ un sitio de comercio electrómco. Aprenderá a mteractuar con el servid~ •. interactuar con la red, a generar imágenes, a manipular fechas Y horu variables de sesión.
- Parte V. Coaatruir proyectos pricticoa CDD PHP y MySQL: Trata ——as del mundo real como la gestión de proyectos de gran tamai\o y funao~es de depuración. Se incluyen proyectos de ejemplo que muestran el potencial Y la versatilidad de PHP y MySQL.
- Parte VI. Apmdices: Incluye varios apctndices que tratan la descripción d! la instalación de PHP y MySQL, una serie de recursos en la Web y el contentdo del CD-ROM del libro.

Conclusión

que disfrute de este libro y que se lo pase tan b~~prendielido-~ utilizar PHP y MySQL como lo hicimOS nosotros cuando empezamos a u tilizar

EL_HOLISTICO@HOTMAIL.COM - 2006

estos productos. Es un placer trabajar con ellos. En poco tiempo, podrá unirse a los miles de desarrolladores que utilizan potentes y robustas herramientas para desarrollar sitios Web din4micos y en tiempo real.

Parte 1

Utilizar PHP

EL_HOLISTICO@HOTMAIL.COM - 2006

1

Curso acelerado de PHP

En este capítulo se realiza un rápido repaso de la sintaxis y de las estructuras de lenguaje de PHP. Si ya programa en PHP, puede utilizar este capítulo para colmar sus lagunas de conocimientos. Si tiene experiencia en programación con C, ASP u otro lenguaje de programación, este capítulo le ayudará a poners_e al día rápidamente.

En este libro aprenderá a utilizar PHP a través de una serie de ejemplos del mundo real, tomados de nuestra experiencia en el desarrollo de sitios de comercio electrónico. Por regla general, los libros sobre programación utilizan ejemplos muy simples para explicar elementos de sintaxis. Nosotros hemos decidido no seguir esta práctica. Nos hemos dado cuenta de que lo que el lector quiere con frecuencia es poder ver un sitio ya creado y operativo, para entender cómo se utiliza el lenguaje, en lugar de otra guía de sintaxis y funciones sin mayor interés que un manual en línea.

Pruebe los ejemplos, introdúzcalos o cárguelos desde el CD-ROM, modiffquelos, divídalos en partes y aprenda a unirlas de nuevo.

En este capítulo comenzaremos por presentar un ejemplo de un formulario.de pedidos de productos en línea para aprender a utilizar las variables, operadores y expresiones en PHP. También analizaremos los tipos de variables y el orden de precedencia de los operadores. Le enseñaremos a acceder a variables de fonnulario y a manipularlas para calcular los totales y los impuestos correspondientes en el pedido de u un diente.

Seguidamente, desarrollaremos un ejemplo de formulario de pedido en línea utilizando nuestra secuencia de comandos de PHP para validar los datos. Examina-

EL_HOLISTICO@HOTMAIL.COM - 2006

remos el concepto de valores Booleanos y presentaremos ejemplos de uso de 1 f, else, del operador ?: y de la instrucción switch. Por último, analizaremos el tema de los bucles para lo cual esaibiremos código de PHP con el que generar tablas HTML repetitivas. Entre los temas clave que se analizarút estmt los siguientes:

- Incrustar PHP en HTML
- Agregar contenido dinúnico
- Acceder a variables de formulario
- Identificadores
- Crea variables declaradas por el usuario
- Examinar tipos de variables
- Asignar valores a variables
- Declarar y utilizar constantes
- Ámbito de variables
- Operadores y su prioridad
- Evaluar expresiones
- Utilizar funciones de variables
- Toma de decisiones con if, else y switch
- Iteración: bucleswhile, do y for

Utilizar PHP

Para realizar los ejemplos de ese capítulo, así como los del resto del libro, necesitará disponer de acceso a un servidor Web con PHP instalado. Para sacar el máximo partido a los ejemplos y casos prácticos que se presentarút, debería ejecutarlos e intentar modificarlos. Para ello, necesitará un banco de pruebas sobre el que experimentar. Si no tiene PHP en su equipo, deberá instalarlo o pedirle a su administrador del sistema que lo haga en su lugar. En el apéndice A encontrará las instrucciones y en el CD-ROM se incluye lo necesario para instalar PHP en UNIX y Windows NT.

Crear una aplicación de ejemplo: Bob's Auto Parts

Una de las aplicaciones más comunes de cualquier lenguaje de secuencia de comandos de servidor es el procesamiento de formularios HTML. Empezaremos

nuestro estudio de PHP implementando un formulario para Bob's Auto Parts, una compailfa ficticia de repuestos de coches. El código utilizado para este ejemplo se incluye en el directorio correspondiente del CD.

Crear el formulario de pedidos

Por el momento, el programador de HI"ML de la empresa de Bob ha creado un formulario de pedido para los repuestos que vende la empresa. En la figura 1.1 se ilustra el aspecto del formulario. Resulta bastante sencillo y es muy parecido a los que se pueden encontrar en Internet. Lo primero que Bob quiere saber es qui pide el cliente, obtener el total del pedido e incluir los impuestos que se deben pagar.

figura 1.1. El formularlo inicial de pedido de Bob sólo registra productos y canUdades.

En el listado 1.1 se recoge parte del código HTML asociado a este formulario.

LI9áldo 1.1. orderform.html. C6dlgo HTML c:onesi,ondiente al formulario bésico de pedido de Bob.

```
cfor,a  action••proc•••order.php•    ..thocl•po•t>
ctable  border•O>
ctr  bgcolor•lccecc:c>
   ctd  width•l50>Itemc/td>
   ctd  width-15>Quentityc/td>
c/tr>
ct.r>
   ctd>Tire•c/tcb
   ctd align••center">cinput     type••text•   114118•"tireqty•    •iae•"l"
       ..xlength•°l"></td>
c/tr>
ct.r>
   ctd>Oilc/td>
   ctd align••center°>cinput    type••text•   Dmlll•"Oilqty•   size•"l"
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
ll&Xleagth•"J">e/tcb
c/tr>
cC.r>
   ctd>Spark  Plugac/td>
   ctd align••caatar">cillput   typa••taxt•   nalle•"•parkqty•   aize••J•
      11AXle119th-"l">c/td>
c/tr>
cc.r»
   ctd col8PIID•"2"   aliga••ceater">cinput   type•"•l&bllit"   •alue•"Subllit
Order• >c/td>
c/tr>
e/table>
c/fora>
```

En primer lugar, hemos establecido la acción del formulario en el nombre de la secuencia de comandos PHP que procesui el pedido del cliente. (Escribiremos esta secuencia de comandos a continuación.) Por regla general, el valor del atributo action es el URL que se cargar6 cuando el usuario pulse el botón de envío. Los datos indicados por el usuario en el formulario se enviarm a este URL a través del mt!todo especificado en el atributo method, ya sea get (ajuntado al final del URL) o po• t (enviado como un paquete diferente).

En segundo lugar, debería fijarse en los nombres de los campos del formulario: tireqty, oilqty y aparkqty. Utilizaremos estos nombres de nuevo en nuestra secuencia de comandos de PHP. Por ello, resulta importante asignar nombres significativos a los campos de formulario que sean sencillos de recordar al escribir la secuencia de comandos de PHP.

Algunos editores HTML generan nombres de campo del tipocampo23 de manera predeterminada. Estos campos resultan difíciles de recordar. Su vida como programador de~ resul~~.l:l.:'~l!_0 ll\~~.!! _si_~t~ 11omb~~re~~!l,ll1~.!i~c,s que se introducen en el campo.

Es aconsejable adoptar un est6ndar de codificación para nombres de campos de manera que todos los nombres del sitio utilicen el mismo formato. De esta forma le resultaril mú sencillo recordar, por ejemplo, si abrevió una palabra en el nombre de un campo o si utilizó guiones bajos para representar espacios.

Procesar el formulario

Para procesar el formulario, tendremos que crear la secuencia de comandos mencionada en el atnbuto action de la etiqueta forlll llamada proceHo:cder. php. Abra su editor de texto y aee este archivo. Escriba el siguiente código:

```
cheacb
   ctitle>Botl'•   Auto  Parta·   Ordar  3eaultac/title>

ch1>1ob'a  Auto  Partac/bl>
ch2..cnder  ...ultac/112>
c/bady>
c/htal>
```


Como observará, todo lo que hemos escrito hasta ahora es simpleaSdigo HlML Ha llegado el momento de agregar algo de código PHP a nuestra secuencia de comandos.

Incrustar PHP en HTML

Agregue las siguientes líneas debajo del encabezado ch2 >:

```
c?php
   echo  •cp>Order  procaaaed.c/p>';
?>
```

Guarde el archivo y cilrguelo en el navegador. Para ello, rellene el formulario de Bob y haga clic en el botón Submit Order. El resultado debería parecerse al ilustrado en la figura 1.2.

Figura 1.2. El texto pando a la Inslrucci6n echo de PHPH refleja en el navegador.

Fijeseen cómo se inaust6 el código de PHP que escribimos dentro de un archivo HrML de aspecto normal. Intente visualizar el código fuente en su navegador. Debería mostrarse la siguiente secuencia:

```
chtal>
chead>
   ctitle>Botl'•   Auto  ,arta·   Order  Reaultac/titla>
c/haad>
cl:lody>
ch1>11ab'a  Auto  Pa~ac/hl>
ch2>0rder  Reaaltac/h2>
cp>Order  p:nM;..8ed.f/p>c/body>
c/htab
```

EL_HOLISTICO@HOTMAIL.COM - 2006

No se ve ninguna secuencia de PHP ya que el intérprete de PHP ha recorrido la secuencia d! comandos.y l~ ha s~stituido co~ su resultado. Por lo tanto, se puede generar cócbgo HTML hmpl0 y vllible a partir de PHP desde cualquier navegador. En otras palabras, el navegador del usuario no necesita entender PHP.

Este hecho ilustra de forma resumida el concepto de secuencias de comandos del lado del servidor. El código PHP se ha interpretado y ejecutado en el servidor Web, lo 'l'le difiere de JavaScript y de otras tecnologías de diente que se interpretan dentro del navegador Web del equipo de un usuario.

El código de archivo se compone de cuatro elementos:

- HTML
- Etiquetas PHP
- Instrucciones PHP
- Espacios en blanco

También podemos agregar

- Comentarios

La mayor parte de las lineas de este ejemplo son sencillo código HTML.

Utilizar etiquetas PHP

El código PHP del ejemplo anterior empezaba con <?php y terminaba con ?>. Es similar a lu etiquetas HTML ya que todas empiezan con un símbolo menor que (<) y te——~ up sfm~lo mayoc que (>). Estossfmbolosse denominan etiquetas PHP que mdican al servidor Web dónde empieza el código PHP y dónde termina. Todo el texto incluido entre las etiquetas se interpretará como PHP. El texto que se encuentre fuera de las etiquetas se procesará como HTML normal. Las etiquetas de PHP le permiten escapar de HTML.

Como veremos a continuación, existen diferentes estilos de etiquetas.

Estilos de etiquetas PHP

En PHP existen cuatro estilos diferentes de etiquetas que podemos utilizar. Los siguientes fragmentos de código son equivalentes:

- EstiloXML

```
c?pbp ecbo 'cp..0rdar proceeeed.c/p>';  ?>
```

&te es el tipo de etiqueta que utilizaremos en el m,r0. Se trata del estilo de etiqu-!a preferido que usar con PHP. El administrador del servidor no puede desactivarlo lo que garantiza su disponibilidad para todos los servidores, muy aconsejable para escribir aplicad~nes que se utilicen en diferentes instaladones. Este estilo de etiqueta se puede utilizar con documentos XML (det • inglés Extensible Markup Language, Lenguaje de marcado extensible). Si tiene previsto utilizar XMLen su sitio, deberla utilizar este estilo de etiqueta.

- Estilo corto

```
e? ecbo •cp..0rder proceeeed.c/p>';  ?>
```

Este estilo de etiqueta es el mú simple y sigue el estilo de una instrucción de procesamiento SGML (del ing14sStandard Generalized Markup Language, Lenguaje de marcado generalizado estándar). Para utilizar este tipo de etiqueta (la mú corta), debe habilitar el parámetro ahort_open_tag en el archivo de configuración o compilar PHP para que utilice este tipo de etiquetas. En el apéndice A encontrará mú información sobre su instalación. No se recomienda utilizar este estilo ya que, aunque est4 habilitado de forma predeterminada, los administradores de sistemas lo suelen desactivar porque interfiere con las declaraciones de documentos XML

- Estilo SCRIPT

```
cecript langua9e••pbp' > ecbo 'cp..0rder proce•eed.c/p>'  l. c/ecript>
```

Este estilo de etiqueta es la mú larga y le resultará familiar si c:opoce JavaScript o VBScript. Puede utilizarla si emplea un editor HTMLque df_problemascon • otros estilos de etiquetas.

- Estilo ASP

```
el ecbo '<P>Order proc••••d.c/p>';  I>
```

- Este estilo.de etiqueta.es-el-41Ue-utilizan-lu P„ginas actiVás-de-servidor-fASPr- o ASP.NET. Se puede utilizar si ha activado el panmetro asp_tags. Puede recurrir a este estilo de etiquetas si utiliza un editor diseñado para ASP.~\ ASP.NET, o si ya ha programado en ASP. De forma predeterminada aparece, deshabilitado.

Instrucciones de PHP

Para indicarle al int*J>rete de PHP qué hacer, se introducen instrucciones de PHP entre lu etiquetas de cierre y lu etiquetas de apertura. En este ejemplo, sólo utilizamos un tipo de instrucción:

```
ecbo •cp..order proc•••ed.c/p>';
```

Como habrá supuesto, la función de la instrucción echo es muy sencilla: imprime (o repite) la cadena pasada en el navegador. Si examina la figura 1.2, el resultado devuelto es la cadena "Order proceased. • en el ventana del navegador.

Como observará, al final de la instrucción echo aparece un pw1\o y comL Este signo se utiliza para ieparar instrucciones en PHP como si se tratara de un ~to al..

EL_HOLISTICO@HOTMAIL.COM - 2006

escribir frases en español. Si ha programado antes en C o en Java, estará familiarizado con el uso del punto y coma con dicha función.

(!no de los errores de sintaxis más comunes al programar es olvidarse de introduar los puntos y comas. Sin embargo, resulta igualmente sencillo descubrir el error y couegirlo.

Espacios en blanco

~ caracteres de espaciado como las líneas nuevas (momos del carro), los espacios y los tabuladores se conocen como espacios en blanco. Como probablemente sepa, los navegadores ignoran los espacios en blancÓ en HTML Y otro tanto hace el motor de PHP. Examine los siguientes dos fragmentos de HTML:

```
chl>Welcome to Bob'• Auto Partalc/hl><P>lfhat  would you like to order
today?c/p>
```

y

```
chl>Welcome      to Bob'•
Auto Parta!c/hl>
cp>What vould you like
' to order today?c/p>
```

Estos dos fragmentos de HTML devuelven el mismo resultado al representarlos • ~ aw,egad~r: ~in embargo, conviene ~zar espacios en Dliinc:o eri.'HTMLpara me,or~r la legibilidad del código. Y otro tanto se puede decir de PHP. El uso de espa~os en b~anco no es ~bligatorio pero facilita la lectura del código si colocamos

-linea s~~~r~~~~i:_~rejent~~~:_

```
echo 'hallo '
ec)lo •vorld
```

y

```
echo 'hello ';echo •world•
```

son equivalentes,pero la primera versión resulta más sencilla de leer.

Comentarios

• Los~o°:'ent.ariosson exactamente eso, comentarios. Su hmción es la de propora~r mdicaaon~ para aqu4llos que leen el código. Los comentarios se pueden u~r par~ ~xphcar el objetivo de la secuencia de comandos, quién la esaibió, por qué se escribió de esa fo~, cuándo se modificó por última vez, etc. Por regla general, todas las !Sf!Cllenaas de comandos de PHP incluyen comentarios, salvo las más sencillas.

~ intérptete de PHP ignorará cualquier texto de un comentario. En concreto, el anaµzador de PHP omite los comentarios que equivalen a espacios en blanco.

PHP admite el uso de comentarios del estilo de C, C++ y secuencias de comandos del núcleo.

A continuación se recoge un comentario del estilo de C multilínea que puede aparecer al inicio de una secuencia de comandos de PHP:

```
/• Autor: lob s.ith
   Modificado por 6lti.. vea: 10 de Abril
   Esta secuencia de comandoa procesa pedidos de clientes.
•/
```

Los comentarios multilfnea deben empezar y acabar con el sfmbolo / •. Como ocurre en C, los comentarios en línea no se pueden anidar.

También se pueden utilizar comentarios de una sola línea, bien en estilo de C++:

```
echo 'cp>Order proce•aed.c/p>';  //Empezara  impriair el pedido
```

o en estilo de secuencia de comandos del núcleo:

```
echo 'cp>Order proce•aed.c/p>'; # llllpezar a impriair el pedido
```

En ambos casos, todo lo que venga después del símbolo de comentario (# o / /) se considera con un comentario hasta el final de la línea y el final de la etiqueta de PHP, según qué ~parezca primero. En la siguiente linea de código, el texto que aparece por delante de la etiqueta de cierre, esto es wr comentario, forma. parte de un comentario. El texto que aparece por detrás de la etiqueta de cierre, esto-no, se cmlSideta,.t fffMt. yirqae se-e1 1Cüetil1afaenrde-la·etiqu~t1n:le'dern!.

```
//esto••  un coaentario ?> e•to no
```

Agregar contenido dinámico

Hasta ahora, no hemos utilizado PHP para hacer algo que no pudiésemos cortiii-; guir con HTML. La razón principal para utilizar un leguaje de secuencia de comandos de servidor es suministrar contenido dinámico a los usuarios de un sitio. Se trata de una aplicación importante porque el contenido que cambia en función de las necesidades del usuario o con el tiempo garantiza la vuelta de los visitantes. PHP nos permite realizar esta tarea de manera sencilla.

Vamos a empezar por un ejemplo sencillo. Sustituya el código de PHP del archivo processorder . php por el siguiente código:

```
c?pllp
   echo •cp>Order proces•ed at '
  ,echo date('B,i,    jS F'!;
   ecbo 'c/p>';
?>
```

En este fragmento, se utiliza la función date () integrada en ~a indic~:~(t cliente la fecha y la ahora a la que procesar el pedido. Esta información será dife~{

te cada vez que se ejecute la secuencia de comandos. En la figura 1.3 se ilustra el resultado de ejecutar la secuencia de comandos en una ocasión:

Figura 1.3. La función date() devuelve una cadena de fecha con formato.

Invocar funciones

,. Examine la llamada a la fundón date (). Se trata de la forma general que adoptan las lla n:"'du de función. PHP incorpora una extensa biblioteca de funciones qu~ puede utilizar para desarrollar aplicacdones Web. La mayor parte de estas funciones pasan y devuelven datos.

-----Examine la siguiente llamada-de-función:-- --.,

date(·e,l. js P'I

Fíjese en que estamos puando una cadena (datos de texto) a la fundón dentro de un par de panmtesis. Esta operación se conoce como llamar al argumento o parámetro de la función. Estos argumentosson los datos de entrada utilizados por la función para devolver resultados específicos.

Utilizar la función date()

La función date◇ es~ que el argumento .que se le pue sea una cadena de formato, que represente el estilo que se desea utilizar para devolver los datos. Cada una de lu letras de la cadena representa una parte de la fecha y la hora. e es la hora en ~ormato 24 horas, i son los minutos precedidos de un cero cuando resulte necesano, j es el día d~l mes sin utilizar un cero inicial, s representa el sufijo ordinal ~ este caso ·~h" del inglés) y Fes el nombre completo del mes. en

(En un capí!:1Jlo posterior se incluye una lista completa de los formatos que admite la f-.mciódate () .)

Acceder a variables de formulario

El único objetivo de utilizar el formulario de pedido es recoger el pedido del cliente. Con PHP resulta muy sencillo obtener los detalles de lo que acaba de introdudr el cliente, pero el método exacto depende de la venión del lenguaje que es~ utilizando y de un parámetro del archivo php. ini.

Variables de formulario

Dentro de una secuencia de comandos de PHP, podemos acceder a cada uno de los campos del formulario como una variable de PHP cuyo nombre se relaciona con el nombre del campo del formulario. En PHP los nombres de variables se reconocen porque comienzan con el signo del dólar ($). Un error habitual es olvidarse de incluir este simbolo. En función de su versión de PHP y de su configuración, hay tres formas de acceder a los datos de un formulario a través de variables. Estos métodos no tienen nombres oficiales por lo que los hemos denominado estilo largo, intermedio y corto. En cada cuo, los campos de formulario de la .página enviada a la secuencia de comandos de PHP se encuentran disponibles en la secuencia de comandos. Puede acceder a los contenidos del tireqty de las siguientes formas:

```
$tireqty                    // Htilo corto
$_POSTl'tireqcy•J           // ••tilo inteE!Dedio        ~.
$HTTP_IOST_VIJUl['tireqty•J // Htilo largo
```

En este ejemplo, y a lo largo del libro, se utiliza el estilo inteilii!dio (es decir, $_POST(·tireqty ·J) para hacer-referencia-aJaS- variables cie_formulario pero puede crear veniones cortas de las variables para facilitar su uso. (Se trata del enfoque recomendado desde la versión 4.2.0 de PHP.)

En su código, puede utilizar otro enfoque, pero para ello debería conocer las diferentes opciones.

En resumen:

- El estilo corto ($tireqty) es práctico, pero requiere activar el parámetro de configuración register_globals. (El valor predeterminado varia de una versión a otra de PHP.) En todas lu versiones desde la 4.2.0 se desactiva de forma predeterminada. Antes, aparecía habilitado y casi todos los programadores lo utilizaban. Este cambio ha provocado gran confusión desde su implementación. En este estilo resulta sencillo cometer errores que podrían convertir a su código en inseguro, por lo que no es una opción recomendada.
- El estilo intermedio ($_POSTl·tireqty ·J) se ha convertido en el enfoque recomendado. Es bastante práctico, pero hizo su aparición en la venión PHP 4.1.0, por lo que no funcionará en lu versiones antigua.
- El estilo largp ($BTl'P_POST_VARS('tireqty•I) es el que incl~ye más detalles. Sin embargo, tenga en cuenta que se considera obsoleto y que es

EL_HOLISTICO@HOTMAIL.COM - 2006

probable que con el tiempo acabe desapareciendo. Solía ser el estilo ms portátil pero se puede desactivar a través de la directiva de configuración register_long_arrays,lo que mejora el rendimiento.

En el estilo corto, los nombres de las variables utilizadas en la secuencia de comandos son iguales a los nombres de los campos del formulario HTML No es necesario declarar las variables en la secuencia de comandos o adoptar ninguna medida para crear dichas variables. Se pasan en la secuencia de comandos básicamente de la misma forma que se pasan argumentos a una función. Si ha escogido este estilo, puede utilizar una variable como $tireqty. El campo tireqty del formulario crea la variable $tireqty en la secuencia de.comandos de procesamiento. Resulta pdctico disponer de este tipo de acceso a las variables, pero antes de activar el parámetro register _globals conviene conocer las razones que han llevado al equipo de desarrollo de PHP a dejarlo desactivado.

Como hemos dicho, este tipo de acceso a las variables resulta pr6ctico, pero es proclive a cometer errores de programación que podrfan comprometer la seguridad de las secuencias de comandos. Si las variables de formulario se convierten autom6ticamente en variables globales, no existid una diferencia clara entre las variables creadas por nosotros y las variables no fiables proc:edentes directamente del usuario.

Si no asignamoS un valor inicial a nuestras variables, los usuarios de las secuencias de comandos pueden pasar variables y valores en forma de variables de formulário-que .. rnadaÁll CGA-la&-nueslfas, Si--se opte-por utilizar el-estilo-corto para acceder a las variables, es aconsejable asignar un valor inicial a las variables.

El estilo intermedio implica la recuperación de variables de formulario de una de las matrices $_POST,. $_GET y $_REQUEST. Una de las matrices $_GBT o $...P.OST contendr6 los detalles-de todas las variables de formulario; t:a matñz ütili; zada depended del método utilizado para enviar el formulario, POSTo GBT, respectivamente. Adem6s, todos los datos enviados a través de los métodos POST y GBT estarán disponibles a tra,·és de $_RBQOEST.

Si el formulario se envía por el método POST, los datos introducidos en el cuadro t ireqt y se almacenarlln en $_POST[• t ireqty • J • Si el formulario se envía utilizando GBT, los datos se almacenar6n en$_GBT[• tireqty • J. En cualquiera de los dos cuos, los datos estar6n disponibles en $_RBQUBST[• t ireqty • J •

Estas matrices son algunas de las denominadas matrices superglobales, a las que volveremos al examinar el 6mbito de las variables.

Si est6 utilizando una versión antigua de PHP, puede que no disponga de acceso a $ POST o$ GET. En las ,·ersiones anteriores a la 4.1.0, esta información se almacenaba enlas matrices $BTTP_POST_VARS y $HTTP_GBT_VARS. Este estilo es el que llamaremos estilo largo y se puede utilizar con versiones antiguas y nuevas de PHP, pero ha sido considerado como obso~ por lo que es probable que no funcione en las veniones futuras. En este estilo no existe un equivalente de $ RBQOBST.

- Si utiliza el estilo largo, puede acceder a la respuesta del usuario a través de $HTTP_POST_VARS('tireqty•] o $BTTP_GBT_VARS['tireqty• J.

tilizad 09 este libro se han probado con la versión S.0 de PHP, Los e,emp os l:lbl ences resulten incompatibJes con versiones anteriores a por lo que es pos1 e que a v amos ue utilice la versión actual. la 4.1.0. Por lo ~to, l~ recr"t nom\resde variable resu]tanun tanto pesados Veamos otro e,e?'p os en un ti de variables conocidas como en el estilo largo e mtermed10, y se basanhas ~tulo posterior comenzaremos matrices, que no se analizar6n de ~eno ~~cap , resulten sencillas de utilizar. por crear copias que bl otra puede utilizar el operador de as1g- Para copiar el valor de UJ '8 va?a e en La'si ente linea de código crea una nación, que en PHP es el 51gno igual(•). IC. contenidos de $HTTP POST nueva variable denominada $tireqty y ~111. VARS [, tireqty, J dentro de la nueva vanable.

```
scireqcy.  $KTTP_POSr_VA1tS['cireqcy'l1
```

cód. l . . . de la secuencia de comandos de Coloque el siguiente bloque de . ig:: =dos de este libro que procesen procesamiento. El resto de secuenaas 51ilar al inicio Como no se datos desde un formulario ~tendr6n ~ bl(ue encima o po~ debajo de la generan ningún r~ultado'. no.1D1porta1 seco : ue se inicie la p4gma:En etiqueta chtml> ru de otras etiquetas\ de~. a . de ~ secuencia de comandos nuestro caso, solemos colocar este ~ oque lnlao para asegurarnos de que resulta f6al de encontrar.

```
c?plap
   //cree DC111breede  variable• corco•
   $Cireqcy.  $BTTP_I08T_VIIIU1('Cireq~y·J,
   $oUqcy • SKTTP_JOST_VUS['oilqcy )i
   Saparlrqcy • SIITTf_POST_VMSI'•P•rkqCy l'
```

bl S i t Soilqty y S•parkqty, y~~ El código crea ~es nuevas vana es t. req y, avésdelmétodoPOSTdesde~~ configura para que contengan los da~osenviados a tr

for111ulario. d dos haga algo visible, agregue las siguientes Para que la secuencia e com~ de PHP: líneas a la parte final de la secuencia de comandos

```
ecllo 'cp>Your ord9r l• •• follov•• c/p>';
ecllo $Cireqcy.• cirescbr />'1
 ello Soilqcy.'  bocel•• of oilcbr />'1
:ello S•parkqcy.•  •park plugecbr />'1
```

obad o el contenido de las variables para ase- Por el momento, no ~emos co!"prd ectos en los campos del formulario. gurarnos de que se han mtroduado atos corr é sucede una vez leido el resto Pruebe a introducir datos ~coaect= ~güner q':nétodo d~ validación de datos a del capítulo, puede que le U\~ esta secuencia de comandos. d r el resultado de la secuencia de Si carga ahora este archivo en el :;• ~4 Los valores reales se mostrann comandos se pareced al ilustrado en ~ en función de lo que escriba en el formulario.

EL_HOLISTICO@HOTMAIL.COM - 2006

Figura 1.4. La Yllrlabla de fonnul8rto Introducida par et USUUO INUltlln de Nicil acceso .,. PIOCNIOrder.php.

En las siguientes secciones se comentan un par de elementos interesantessobre este ejemplo.

Concatenar cadenas

En la secuencia de comandos, utilizamos echo para imprimir el valor introducido por el usuario en cada uno de los campos de formulario, seguido por texto explicativo. Si examina de cerca las instrucciones echo, verá que el nombre de la variable y que el texto que sigue están separados por un punto (.), como se puede ver a continuación:

```
ecbo $tir9qey.• tire•cbr />';
```

t~te es el operador de c~ncatenación de cadenas y se utiliza para unir cadenas (secctones de texto). Lo utilizañ con frecuencia con echo para mostrar información en pantalla. Se suele recurrir ı cH para no tener que escribir varios comandos echo.

Para las variables que no sean matrices, puede encerrarlas entre comillas dobles para su representaciónen el navegador. (El estudio de las matrices se aborda en un capitulo posterior.) Por ejemplo:

```
Niio "$tireqty tire•cbr />";
```

Esta secuencia es equivalente a la primera. Los dos formatos son válidos y la elección de uno u otro es una cuestión personal. El proceso de sustituir una variable con sus contenidos dentro de una cadena se denomina interpolación.

~ interpolación es exclusiva de las cadenas entre comillas dobles. No se puede dadir nombres de variables dentro de una cadena entre comillas simples de esta forma. Si se ejecuta la siguiente línea de c6digo:

```
ecbo '$tinqty tire•cbr />';
```


enviará la secuencia "$tireqty tireecbr /:." al navegador. Si se utiliun -- comillas dobles, el nombre de la variable se sustituid por su valor. Si se utilizan comillas simples se envian el nombre de la variable, o cualquier otro texto, sin alterar.

Variables y literales

Las variables y las cadenas concatenadas en cada una de las instrucciones echo son cosas diferentes. Las variables son un símbolo pua sustituir a los datos. Las cadenas son datos en sí mismas. Cuando se utiliza un segmento de datos como tal en un programa como éste, se designa como literal pua distinguirlo de las_variables. $tireqty es una variable, un súnbolo que representa los datos escritos por el cliente. Por el contrario, ' tireecbr /:.' es un literal. Se puede tomar por su valor literal Bueno, casi. ¿Recuerda el segundo ejemplo anterior? PHP sustituye el nombre de la variable $t ireqty en la cadena por el valor almacenado en la variable.

Recuerde que existen dos tipos de cadenas en PHP, unas encerradas entre coffl:i· Uu dobles y otras entre comillas simples. PHP intentará evaluar 1-.:Pdenas con comillas dobles, lo que dará como resultado el comportamiento visto *leriormente. Las cadenas con comillas simples se tratarán como verdaderos litsales.

Recientemente ha aparecido una tercera forma de especificar cadmas. La sintaxis heredoc (<<<), que a los usuarios de Peri le resultará familiui,ae aftadió a PHP4. Esta sintaxis permite especificar extensas cadenas de forma 91tp11te, meidiante un man:ador final que se utilizará para finalizar la ~ siguiente ejemplo crea una cadena de tres lineas y la reproduce:

```
ecbo ccctbe&lld
    Uual
    u .... 2
    11Da• 3
tlleBDd
```

El elemento theEnd es totalmente arbitrario. Simplemente~ debe aparecer en~/._ el texto. Para cerrar una cadena heredoc debe incluir un elemento de cierre 11 comienzo de la linea. Lu cadenas heredoc se interpolan como las cadenas entre comillas dobles.

Identificadores

Los identificadores son nCllllbres de variables. (Los nombres de funciones y clases son también identificadores. Los nombres y las clases se examinarán en capítulos posteriores.) Los identificadores siguen algunas reglas sendllas: • .. \)

- Loa identificad~res puede tener cualquier longitud y pueden incl~ ~~~ #
  números y guiones bajos. ~.tl:S

EL_HOLISTICO@HOTMAIL.COM - 2006

- Los identificadores no pueden comenzar por un número.
- En PHP, los identificadores discriminan entre mayúsculas y minúsculas. $tireqtynoes lo mismo que$TireQty. Un error común en programaáón consiste en intentar utilizarlos indistintamente. Los nombres de las funáones son una excepáón a esta regla (el uso de mayúsculas y minúsculas es indiferente).
- Una variable puede tener el mismo nombre que una función. Este hecho puede dar lugar a confusiones por lo que conviene evitarlos. Así mismo, no se pueden crear dos funáones con el mismo nombre.

Crear variables declaradas por el usuario

Puede declarar y utilizar sus propias variables además-de las variables que se pasan desde el formulario HTML.

Uno de los rasgos de PHP es que no requiere declarar variables.antes de utilizar las. Las variables se crean al asignarles un valor (viase la siguiente sección para conocer los detalles).

Asignar valores a variables

Pai-a asignar valores a las variables se utiliza el operador de asignaá6n (•) como hicimos al copiar el valor de una variable en otra. En el sitio de Bob, queremos obtener el número total de artlc:ulos pedidos y la cantidad ~1~~.~debe.abonar. Podemos crear ci(M.l variables-pan-almacenar estos valores. En primer lugar, initillizamos cada una de estas variables en cero.

Agregue las siguientes líneas al final de la secuencia de comandos de PHP:

```
Stotalqty • 0;
Stotala.ount • 0.00;
```

Cada una de estas lfneas crea una variable y le asigna un valor literal. También puede asignar valores de variables a las variables, por ejemplo:

```
Stotalqty • 0;
Stotala.ount • Stotalqty;
```

Tipos de variables

Un tipo de variable hace referencia al tipo de datos que se almacenan en ella. PHP cuenta con un completo conjunto de tipos de datos. En los distintos tipos de datos se pueden almacenar diferentes datos.

Tipos de datos de PHP

PHP admite los siguientes tipos de datos

- EnterO: Ctilizado para números enteros
- Jllc,laJlte (o Doble): Utilizado para números reales
- Cadena= Utilizado para cadenas de caracteres
- Boo)eu,O: Utilizado para valores verdaderos o falsos
- Matriz: Utilizado para almacenar con1untos e d datos del mismo tipo (viese un capitulo posterior)
- ObjelO: Utilizado para almacenar mstanaas de clases (véase un capitulo posterior)

También existen dos tipos espeaaúl d les: NULL y de recurso. Las variables a las que nose les ha a5ion.ado un valor, no es finidas o se les ha asignado el valor lfYLL son de tipo NUL~ e

Algunas tunaones aneo rporadas (como las funciones de base de datos) d~uel ven variables con tipo de recurso. Representan recursos externos (como conexiones de base de ~abtc;5~ Esperomut=ones robable que necesite manipular directamente este tipodevan las s1,1el~ 4e.Yc;,lv.tr y dtbm ~ ~ parámetro& a otras funaoaes.

Control.de tipos

PHP es un le~guaJe con ":'~::!i!eri:' ti mu débil. En la mayoria de lo11l: lenguajes, las vanabl~:61:fer tipo~e datos y dicho tipo de da~ debe declararse antes utilizar la variable, con ocurre en C. En PHP' el tipo de !8~le d el valor que se le asigne. coreiemp VIl.!1~...~,, y Stotalamount, deben determinarse sus tipos iniciales, de la siguiente forma:

```
$totalqt;Y • 01
$totaia-auat • o.001
```

Como hemos asagna do o un valor entero, a $totalqty, se tratar A de una vana- ble de tipo entero. De maneuedra SI 'milar $totalallOUDt es de tipo doble. Pero por extrai\o que p 'er podemos agregar una linea a nuestra se- cuencia'de comandos de la siguiente forma:

```
Stotal•- e • ·M1lo';
```

La variable $tota~aaount ~ado ráentoncesdenpoca d ena. PHP cam aa b el tipo de en ella en cualquier momento dado. la variable en fundón del valor

EL_HOLISTICO@HOTMAIL.COM - 2006

Esta capacidad para cambiar los tipos de manera transparente al inatante puede resultar extremadamente útil. Recuerde que PHP sabe automáticamente• quf tipo de datos se ailade a una variable y devolverá los datos con el mismo tipo tras recuperarlos de la misma.

Convertir tipos

Puede simular que un tipo de variable o valor sea de un tipo diferente utilizando una conversión de tipo. Este recurso funciona de la misma forma que en C. Basta con colocar el tipo temporal entre corchetes delante de ta variable que se desea convertir. Por ejemplo, podríamos haber declarado las dos variables anteriores utilizando una conversión.

```
$totalqty • 0;
$totalamount • (double1$totalqty;
```

La segunda lfnea indica "toma el valor almacenado en $totalqty, interpn!talo como doble y almacfnalo como $totalamount". La variable $totalamount será de tipo doble. La variable de conversión no cambia de tipo por lo que $totalqty seguirá siendo de tipo entero.

Variables de tipo variable

PHP proporciona otro tipo de variable: la variable de tipo variable. Las variables de tipo variable permiten cambiar el nombre de una variable dinámicamente.

(Como puede observar, PHP proporciona una gran libertad en este sentido: todos los lenguajes permiten cambiar el valor de la variable, pero no hay muchos que permitan cambiar el tipo de variable y menos aún que permitan cambiar su nombre.) Su funcionamiento consiste en utilizar el valor de una variable como nombre de otra. Por ejemplo, podemos establecer

```
$va~ • 'tireqty•:
```

A continuación podemos utilizar $$varname en lugar de Stireqty. Por ejemplo, podemos establecer el valor de $tireqty:

```
$$va~• 5:
```

Esto equivale a

```
Stireqty • s,
```

Puede que parezca un poco confuso, pero no se preocupe porque volveremos sobre su uso posteriormente. En lugar de tener que enumerar y utilizar cada variable de formulario por separado, podemos utilizar un bucle y una variable para procesarlas todas automáticamente. En la sección dedicada a los bucles for se incluye un ejemplo que ilustra este hecho.

Declarar y utilizar constantes

Como vimos anteriormente, podemos cambiar el valor almacenado en una variable. Talllbifn podemos declarar constantes. Una constante almacena un valor como una variable con la diferencia de que se establece una vez y no se puede cambiar en ningún otro punto de la secuencia de comandos.

En nuestra aplicación de ejemplo, podríamos almacenar los precioe de los artfculos de venta en forma de constantes. Para definir constantes puede utilizar la función define:

```
define (•TIUl'RICB '• 1001;
define('OIIJIRICS', 101,
define('SPARXPaICB', 41;
```

Agregue estas líneas de código a la secuencia de comandos. Ahora dispone de tres constantes que puede utilizar para calcular el total del pedido de un cliente.

Los nombres de las constantes utilizan siempre mayúsc:ulas. Se trata de una convención tomada de C que facilita su distinción de las variables. Esta convención no es obligatoria pero contribuye a que la lectura y el manteniaiie)o del código resulten más sencillos.

Tenemos tres constantes que se pueden utilizar para calcular el • del pedido • del cliente.

Una diferencia importante entre las constantes y las variables: que cuando hacemos referenda a una consfailte, no lleva antepuesto el sfmbo el dólar. Si desea utilizar el valor de una constante, utilice únicamente su nom or ejemplo, para u til~I'.~ de..!~~~~~ qu4!~ca~~ "e~, podega~~¡r,

```
echo TIRBftICB,
```

Además de las constantes definidas, PHP establece un gran número de constantes propias. Una forma sencilla de verlas consiste en ejecutar el comandopbpinfo ():

```
phpinfo(l I
```

Este comando devuelve una lista de variables y constantes predefinidas de PHP, además de otra información útil. lremoe comentándola a medida que vayamos avanzando. Otn diferencia entre variables y constantes es que éstas sólo pueden almacenar datos de tipo Booleano, entero, flotante o de cadena. Estos tipos se conocen como valores escalares.

Ámbito de variables

El término ámbito hace referencia a los lugares dentro de las secuencias de comandos en los que resulta visible una variable dada. A continuación se desc:riberi;: las seis reglas que se aplican a los ámbitos de PHP:

EL_HOLISTICO@HOTMAIL.COM - 2006

- Las variables superglobales incorporadas resultan siempre visibles dentro de una secuencia de comandos.
- Las constantes, una vez declaradas, siempre resultan visibles de forma global, y se pueden utilizar dentro y fuera de una función.
- Las variables globales declaradas en una secuencia de comandos resultan visibles a lo largo de la secuencia de comandos pero no dentro de las funciones.
- Las variables utilizadas dentro de funciones que se declaran como globales hacen referencia a la variable global del mismo nombre.
- Las variables creadas dentro de funciones y declaradas como estáticas son invisibles desde el exterior de la función pero conservan su valor entre las ejecuciones de ésta, como veremos en un capítulo posterior.
- Las variables creadas dentro de funciones tienen restringido su ámbito a la función y dejan de existir cuando ésta desaparece.

A partir de la versión 4.2 de PHP, las matrices $_GETy $_POSTasí como otras variables especiales llevan asignadas sus propias reglas de ámbito. Éstas se conocen como superglobables y se pueden ver en todas partes, tanto dentro como fuera de las funciones.

A coatinuación se recoge la lista completa de variables globales·

- $GLOBALS, una matriz con todas las variables globales. (Al igual que la palabra clave global, nos permite acceder a las variables globales dentro de una función, por ejemt:>lo_co ~~ ~GLO~~! c · ~':':1:i~1.e~.!)
- $_SERVBR, ~~~triz con las variables de entorno del servidor
- $ GET, una matriz con las variables pasadas a la secuencia de comandos a través del método GET
- $ POST, una matriz con las variables pasadas a la secuencia de comandos a través del método POST
- $_COOKIE, una matriz con las variables de cookies
- $_FILES, una matriz con las variables relacionadas con las cargas de archivos
- $_ENV, una matriz de variables de entorno
- $ REQOEST, una matriz con todas las variables de entrada de usuario, incluy~do los contenidos de entrada de $_GET, $_POSTy $_COOKIE
- :$_SESSION,una matriz con las variables de sesión

Volveremos sobre cada uno de estos tipos a medida que vayamos avanzando. Analizaremos el ámbito de las variables al estudiar las funciones. Por el momento, todas las variables que utilicemos serán globales de manera predeterminada.

Utilizar operadores

Los operadores son símbolos que se pueden utilizar para manipular valores y variables realizando una operación sobre ellos. Tendremos que utilizar algunos para calcular los totales y los impuestos que aplicar al pedido del cliente.

Ya hemos mencionado dos operadores: el operador de asignación (•) y el operador de concatenación (.). A continuación examinaremos la lista completa.

En general, los operadores pueden adoptar dos o tres argumentos (aunque la mayoría toman dos). Por ejemplo, el operador de asignación adoptar dos argumentos: la ubicación de almacenamiento en la parte izquierda del símbolo • y una expresión en la parte derecha. Estos argumentos se conocen como operandos, es decir, los elementos sobre los que se opera.

Operadores aritméticos

Los operadores aritméticos son bastantes claros: se trata de los operadores m~-temáticos de uso más común. En la tabla 1.1 se recogen los operadores aritméticos.

Tabla 1.1. O~ aritméticos de PHP.

Operado	Nombre	Ejemplo
+	Suma	$a + $b
	Resta	-'b
*	Multiplicación	$a * $b
/	División	$a '/ $b
%	Módulo	$a 1 $b

Con todos estos operadores podemos guardar el resultado de la operación. Por ejemplo:

```
$reeult ·$a+  $b;
```

La suma y la resta funcionan como se esperaría. Estas operaciones suman o restan, respectivamente, los valores almacenados en las variables $a y $b.

También puede utilizar el símbolo de resta (-) como operador unitario (es decir, un operador que toma un argumento u operando) para indicar valores negativos; por ejemplo:

```
Sa • ·1;
```

La multiplicación y la división también funcionan de la forma esperada. T~~ ..... en cuenta que se usa el asterisco como operador de multiplicación en lugar dV~~::-

EL_HOLISTICO@HOTMAIL.COM - 2006

símbolo habifual de multiplicación y la barra inclinada como símbolo de la división. El operador de módulo deyuelve el resto de la división de la variable $a por la variable $b. Considere el siguiente fragmento de código:

```
S• • 2'71
Sb • 101
Srenlt: • Sa1Sbl
```

El valor almacenado en la variable $resultes el resto de dividir 27 por 10, es decir, 7. Tenga en cuenta que los operadores aritméticos se suelen aplicar a las variables enteras o dobles. Si se aplican a cadenas, PHP intentad convertir la cadena en un número. Si contiene una "e" o una "E", la convertir, en una variable doble, de lo contrario la convertir, en un entero. PHP busca números al principio de la cadena y los utiliza como valores (si no hubiera ninguno, el valor de la cadena seria cero).

Operadores de cadena

Ya hemos visto y utilizado el único operador de cadena. Puede utilizar el operador de concatenación de cadenas para agregar dos cadenas, y que genere y almaeene un resultado como haríamos si utilizúamos el operador de suma para sumar dos números.

```
$a- •Bob•• •;
Sb • •Auto tazt• •;
Sre•ult: • $a.$b;
```

La variable $result contendr, la cadena •aob•s Auto Parts•.

Operadores de asignación

Ya hemos visto el operador de asignación b'nco •. A este operador se har, siempre referencia como operador de asignación y se lee como "establecer en•. Por ejemplo:

```
$t:ot:alqt:y • O:
```

Esta secuencia se lee como •$totalqty se establece en cero". Ya explicaremos por qué al hablar de los operadores de comparación en una sección posterior.

Oeyolver valores de asignación

E! uso del operador de asignación devuelve siempre un valor general similar a otros operadores. Si escribe

```
Sa • Sb
```

el valor de la expresión es el resultado de agregar las variables $a y $b.

~ manera amloga, puede escribir:

```
sa • o,
```

El valor de toda la expresión es cero.
Esta secuencia permite realizar operaciones como:

```
Sb • 6 + l$a • 5ll
```

Esta secuencia establece el valor de la variable $ben 11. Por regla general, el valor de ~ la instrucción de asignación es el valor que se asigna en el operando de la izquierda. Al calcular el valor de una expresión, se pueden utilizar los padntesis para asignar prioridad a una subexpresión como hemos hecho en el ejemplo. Su funcionamientoes el mismo que en ma~tic:as.

Combinar operadores de asignación

Además de la asignación iunple, existe un conjlD\tode operadores de asignación combinados. Se trata de formas abreviadas de realizar otra operación sobre una variable y de asignarle el resultado. Por ejemplo:

```
•• +• s,
```

Esta secuencia equivale a escribir:

```
S• • Se + 5;
```

Existen operadores de asignación combinados para cada uno de los pradores aritméticos así como para el operador de concatenación de cadenas. En la tabla 1.2 se recoge un n!SUDlen.de todos los..operadores de..asignación..combiaados }LSU-.-.-.- efecto.

Operado·	Uso	Equiva·->
''	$a '' $b	$a• $a+ $b
	$a $b	$• • $a - $b
''	$a '' $b	$• •$a• $b
/•	$a /• $b	$a • $a/ $b
••	$a •· $b	$•. $a I $b
''	$a '' $b	$•. $a. $b

Incremento y decremento previo y posterior

Los operadores de incremento(++) y decremento(--) previa y poeterior son ~f. similares a los operadoies +•y-•, pero con un par de variaciones.

EL_HOLISTICO@HOTMAIL.COM - 2006

Los operadores de incremento presentan dos efectos: inaementan un valor y lo asignan. Considere la siguiente secuencia:

```
$&•4;
echo ++$al
```

La segunda línea utiliza el operador de incremento previo, denominado así por-,ue ++ aparece delante ~e $a. Este operador tiene dos efectos: en primer lugar mcrementa $a en~ urudad y en segundo lugar devuelve·el valor inaementado. En este caso, $a se 11\Crementa en 5 y se devuelve e imprime dicho valor. El resultado de toda la expresión es *S.* (Ffjese en que el valor que se almacena en realidad en $as~ ha variado: ?º estamos devolviendo simplemente '$a + 1.)

Sm emba~o, s1 colocamos_++ después de $a, utilizamos el operador de incremento postenor. Su efecto es diferente al anterior. Considere la siguiente secuencia:

```
$a•4:
echo $a++;
```

En este caso, los efectos se invierten. En primer lugar, se devuelve e imprime el valor de $a y en segundo_lug~ ~ ~ementa. El resultado de toda la expresión es ~- &te ~ el valo~ que se impnaurá. Sm embargo, el valor de $a tras ejecutar esta ~~cc1ón será e, Como probablem~te habrá adivinado, el comportamiento es similar para el operador - -, con la diferencia de que el valor de $a se reduce en lugar de incrementarse.

Referencias

~~ interesante utilidad ~ el o~erador de referencia,&, que se utiliza en combinaaón con. el operador. de as1gnaaón. Por-regla-general, ruando una-variable se ~igna a otra, se realiza una copia de la primera y se almacena en memoria. Por e.,emplo:

```
$&. 5;
$b • $a,
```

Estas líneas de código realizan una segunda copia del valor almacenado en $a y lo ~ardan en $b. Si cambiamos el valor de $a en un momento posterior, $b no vanará:

```
a• 7; // $b aeguirA aiendo 5
```

Para evitar que se haga la copia y que se guarde en memoria puede utilizar el operador de referencia, &. Por ejemplo:

```
$& • 5;
$b • •S•1
$• • 71 f·l $a y $b aon ahora 7 i.. do•
```

Las referencias pueden resultar un tanto complejas. Recuerde que una referencia es como un alias, no como un puntero.

Tanto $a como $b apuntan a la misma información. Puede cambiarlo si anula la configuración de una de ellas:

```
1111Ht($&);
```

Al realizar esta operación no se modifica el valor de $b (7) pero anula el vínculo entre $a y el valor *1* almacenado en memoria.

Operadores de comparación

Los operadores de comparación se utilizan para comparar dos valores. Las expresiones que utilizan estos operadores devuelven el valor 16gico true o el valor lógico false en función del resultado de la comparación.

El operador iguales

El operador de comparación iguales, •• (dos signos iguales), permite determinar si dos valores son iguales. Por ejemplo, podemos utilizar la expresión

```
$a •• $b
```

para determinar si los valores almacenados en $a y en $bson iguales. El resultado devuelto por esta expresión será true si son iguales o false si no lo son.

"ReSultnencdto confu11dh este ope1ador con el opereder de aaignaaGR{-). &&a confusión no generará un error pero impedirá que se obtengan los valores deseados. En general, los valores distintos a cero se evalúan como true y los valores iguales a cero se evalúan como false. Suponga que hemos inicializado dos variablescfe lasiguiente forma:

```
$a • 5;
$b • 7;
```

Si prueba $a • $b, el resultado será true. ¿Por qué? El valor de $a " $bes el valor uignado a la parte izquierda de la expresión, que en este caso es 7. Se trata de un valor distinto a cero, por lo que la expresión se evalúa como true. Si su intend6n es probar $a •• $b, que devuelve false, habrá introducido un error de lógica en su código que puede resultar extremadamente dificil de detectar. Compruebe siempre el uso de estos dos operadores y verifique si ha utilizado el que tenía previsto.

Se trata de un error muy sencillo de cometer y es muy probable que caiga en él muchas veces a lo largo de su carrera como programador.

Otros operadores de comparación

PHP admite otros operadores de comparación, que se recogen en la tabla 1.3.

Uno de los más destacados es el nuevo operador de identidad, •••, introducido en PHP, que devuelve true sólo si los dos operadores son iguales y del mismo tip~.

EL_HOLISTICO@HOTMAIL.COM - 2006

T•bl• 1.3. Operadores de companicl6n de PHP.

Operado	Nombr	Uso
	Igual	$a -- $b
	ld6ntlco	$a --- $b
!•	Distinto	$a !• $b
!••	Distinto	$a !•• $b
<>	Distinto	$a <> $b
e	Menor que	$a < $b
>	Mayor que	$a > $b
c.	Menor o igual que	$a <• $b
>•	Mayor o igual que	$a >• $b

Operadores lógicos

Los operadores lógicos se utilizan para combinar los resultados de condiciones lógicas. Po~ ejemplo, puede que nos interese que el valor de una variable, $a, se encuentre-entre e y 100. Para ello tendríamos que probar las condiciones $a >• o y $a <• 100, utilizando el operador ANO, como se indica a continuación:

$&->•.a_"" Sa_j: 100

PHP admite el uso de los operadores lógicos ANO, OR, XOR (o exclusivo) y NOT. En la tabla 1.4 se resumen los operadores lógicos y su uso.

T•bl• 1.4. Operadores l6glcos de PHP.

	NOT	!$b	Devuelve true si $b es false y viceversa
IIII	ANO	$a IIII $b	Devuelve true si $a y $b son true; de lo contrario devuelve falH
II	OR	$a II $b	Devuelve true si $a o $b o ambas son true: de lo contrario dewelve fal-
and	ANO	$a ami $b	Igual que "". pero con prioridad mú baja
or	OR	$a or $b	Igual que ||, pero con prioridad mú baja

Los operadores and y or tienen una prioridad inferior a la de los operadores"" y II. En una sección posterior se analizan el tema de la prioridad.

Operadores bit a bit

Los operadores bit a bit permiten tratar un entero como una serie de bits utiliados para representarlos. Estos operado~es no se utilizan demasiado en PHP. En la tabla 1.5 se recoge un resumen de los operadores bit a bit.

Tabi. 1.5. Operadores bit a bit de PHP.

Operador	Nombr	Uso	Resultad
&	[illegible]	[illegible] $b	[illegible] cen en el ntSUltado
	OR bit a bit	$a $b	Los bits asignados en $a o $b se establecen en el resultado
	NOT bita bit	-$a	los bits asignados en $a no N establecen en el resultado y viceversa
	XOR bit a bit	$a A $b	Los bits asignados en $a·o tb paro no en ambos se establecen en _ — _ ultado
<e	Desplazamiento a la izquierda	$a << $b	Mueve $a a la izquierda en función de los bits de $b
>>	Desplazamiento a i. derac:ha	$a >> $b	Mueva $a a la derecha en función de los bits de $b

Otros operadores

Ad~~ de los operadores vistos hasta ahora, existen otros. El operador coma :, (,) se utiliza para separar argumentos de función y otros elementos de lista. Se suele utilizar de manera ocasional.

Los operadores new y -> se utilizan para crear instancias de una clase y para acceder a los miembros de clase, respectivamente. Estos operadores se analizarán detalladamente en un capitulo posterior.

Existen otros tres operadores que se examinan brevemente a continuación.

Operador ternario

Este operador, ? :, funciona de la misma forma que en C:. ~opta la siguiente forma:

```
condici6n? valor •i tru• : valor ai fal••
```

El operador temario es similar a la versión de la expresión de una instrucción if-else, que se an.lizar4 ns adelante en este capitulo.

EL_HOLISTICO@HOTMAIL.COM - 2006

Vemos un sencillo ejemplo:

```
l$9rade >so? 'Aprobado' : 'suapaa.o');
```

Esta expresión evalúa la nota del estudiante como 'Aprobado ' o ' Suspenso '.

Operador de supresión de error

El operador de supresión de error, ~. se puede utilizar por delante de cualquier expresión, es decir, de todos aquellos elementos que generen o tengan un valor. Por ejemplo:

```
••. ecs1101;
```

Sin el operador~. esta linea generará un aviso de división por cero (inténtelo). Si se incluye el operador, se suprimirá el error.

Si tiene previsto suprimir advertencias de esta forma, debería escribir código de control de errores que compruebe cuándo tiene lugar una advertencia. Si ha configurado PHP con la función track_errors activada, el mensaje de error se almacenará en la variable global $php_errormsg.

Operador de ejecución

Enrealidad el operador de ejecución está formado por un par de operadores: dos acentos graves (`). Este símbolo es dilerente a la comilla simple (en el teclado se sitlta en la tecla con el símbolo de difresis).

PHP intentará ejecutar todo lo que se incluya entre estos símbolos como un comando en la línea de comandos del servidor. El valor de la expresión es el resultado4.el c;o~do,

Por ejemplo, en sistemas operativos del tipo UNIX, puede utilizar:

```
$out• 'l• -la';
echo •cpre>'.$out.•c/pre>';
```

O ~l equivalente en un servidor Windows:

```
$out • "dir e:';
ecbo 'cpre>'.$out.•c/pre>';
```

Cualquiera de estas versiones devolverá un listado de directorio *y* lo almacenará en $out. A continuación se puede imprimir en el navegador o manipular de otra forma. Existen otras formas de ejecutar comandos en el servidor, como veremos en un capítulo posterior-

Operadores de matriz

Exiatén diferentes operadores de matriz. Los operadores de elemento de matriz ([)) le permiten acceder a los elementos de una matriz. También puede utilizar el operador •> en determinados casos. Los analizaremos en uno de los siguientes capítulos, aunque se los mostramos a continuación.

Tabla 1.1. Operadores de matriz de PHP.

Operador	Nombre	Uso	Resultado
+	Unión	$& + $b	Devuelve una matriz que contiene todo lo incluido en $a y $b
	Igualdad	$& •• $b	Devuelve true si $a y $b tienen los mlsmoa elementos
	Identidad	$& ••• $b	Dewelve true si $a y $b tienen los mismos elementos en el mismo ORien
	Desigualdad	$& !• $b	Dewelve true si $a y $b no son Iguales
	Desigualdad	$• <> $b	Dewelve true si $a y $b no son iguales
!••	No idéntico	$a' !•• $b	Dewelve true si $a y $b no son idénticos

Observará que los operadores de matriz descritos en la tabla cuentan con operadores equi~alentes que funcionan con variables escalares. Siemp~ que recuerde que + realiza sumas en tipos variables *y* sirve para unir matrices, aunque no le interese la aritmffica de la operación, ambos comportamientos tendrán sentido. No se pueden comparar matrices con tipos escalares.

El operador de tipo

Existe un c,perador de tipo: iz:¡stan<:eof, que se utiliza en la programación orientada a objetos, tema que analizaremos en un capítulo posterior.

El operador Instanceof le peñníte comprobar si Üí\ objeto es üiia-instandade una clase concreta, como se indica en el siguiente ejemplo:

```
el••• llall¡,leC1•••{)ı
$11Y()bject • nav sal1pleCl•••II;
if ($myObjact inecanceof ••n,pleCl•••>
   echo "IIY()bjecc is an in•cance of samplacl..••;
```

Utilizar operadores: calcular los totales de los formularios

Ahora que ya sabemos cómo utilizar operadores de PHP, podemos calcula~ los totales *y* los impuestos en el formulario de pedido de Bob.

Para ello agregue el siguiente código al final de la secuencia de comandos de PHP:

```
$tocalqcy • oı
$cocalqcy • $tireqcy + $oilqcy + $aparkqtyı
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
echo 'Items ordered: '.$totalqey.'cbr   />';

$totala80Ullt • o.oo,

define('TIR&PaICB',  1001,
define( ·OILl'RICB.', 10);
defi11e(ıSPAIUCPRICS',  4);

$totalamoullt • $tireqty • TIR&PRICB
      + soilqty.  OILPRICS
      + $8parkqty • SPAIUCPRICB;

echo •subtotal: $•.11ullber_for.atlStotalamount,l).'cbt  />';

Staxrate.  D.10;  // el i11Pue•to de venta• local •• del 101
stotalamouDt • $total&11CM111t • |1 • $taxratel,
echo 'Total including tax, $'.nWÜler_format($totalamowit,21.'cbr    />';
```

Si actualiza la página en la ventana del navegador, verA un resultado similar al ilustrado en la figura 1.5.

FIguni 1.5. Se han calculado los totales del pedido de dientes, se les ha aplicado formato y se ha mostrado.

Como puede observar, en esta secuencia de código hemos utilizado varios operadores. El operador de suma (+) y el de multiplicación (•) se encargan de calcular lu cantidades y el operador de concatenación (.) establece el resultado en el navegador. También hemos utilizado la función number_format () para aplicar formato a los totales como cadenas con dos decimales. Se trata de una función de la biblioteca de matemáticas de PHP.

Si se detiene a examinar los cálculos, es posible que se pregunte por qué se realizaron en ese orden. Por ejemplo, considere esta instrucd6n:


```
Stotala-t    • $tireqty • TIUHICB
        + $oilqty • OILPRICB
        • $8p&rkqty • SP1JUtPRICB1
```

La cantidad total parece conecta pero, ¿por qué realizar las multiplicaciones antes que las sumas? La respuesta reside en la precedencia de los operadores, es decir, en el orden en el que se calculan.

Precedencia y asociatividad: evaluar expresiones

Por regla general, los operadores se evalúan siguiendo una precedencia u orden de prioridad fijado.

Los operadores llevan asignada una asociatividad, que es el orden en el que se evalúan los operadores con el mismo orden de prioridad. Suele ser de izquierda a derecha (o izquierda simplemente), de derecha a izquierda (o derecha simplemente) o no resulta relevante.

En la tabla 1.7 se recoge la precedencia de operadores y la asodatividad en PHP. En esta tabla, se incluyen los operadores de prioridad más baja en la parte superior y la precedencia va aumentando según se desciende por ella.

Tabla 1.7. Pracedencia de operadoru en PHP.

Asociatividad	Operadores
izquierda	
izquierda	or
izquierda	xor
izquierda	and
derecha	print
izquierda	• +• ·- /• .• ,. &• | - -. -- <<• >>•
izquierda	? :
izquierda	||
izquierda	
izquierda	
izquierda	
izquierda	
nld	
nld	< <= > >=
izquierda	<< >>

EL_HOLISTICO@HOTMAIL.COM - 2006

Asociatividad	Operadores
izquierda	+ ·
iZquiercla	• / t
derecha	·I - ++ ·· (entero) (doble) (cadena) (matriz) (objeto) •
derecha	(]
n/d	new
n/d	()

Ffjese en que el operador con mayor prioridad no se ha analizado todavía: los paréntesis. El efecto de los paréntesis consiste en incrementar la prioridad de todos los elementos incluidos en su interior. Esto símbolos permiten alterar las reglas de prioridad cuando resulte necesario.

Recuerde la siguiente sección del último ejemplo:

```
$total..aunt • $totala11011at • (1 + Staxrate);
```

Si hubiéramos escrito

```
'stotala-t • Stotala-unt • 1 + $taxrate!
```

el operador de multiplicación, que tiene precedencia sobre el operador de suma, se aplicaría en primer lugar, lo que nos daría un resultado incorrecto. El uso de los paréntesis permite forzar el cálculo de la subexpresión 1 + $taxrate en primer lugar: Püede utilizar táritois paréntesis como desee en una -expresión. En primer lugar se calcularin los más internos.

Utilizar funciones de variables

Antes de abandonar el mundo de las variables y de los operadores, vamos a examinar las funciones de variables de PHP. Se trata de una biblioteca de funciones que permite manipular y probar variables de distintas formas.

Probar y establecer tipos de variables

La mayor parte de estas funciones se utilizan para probar el tipo de una variable. Las dos más generales son gettype () y aettype (). &tas tienen los siguientes prototipos de fwlciones; es decir, lo que esperan los argumentos y lo que devuelv~.

```
atri119 pttype(lli:ud var>,
' llool aettype (ai:ud var, atring typel I
```


Para utilizar gettype (), la pasamos en una variable. Determinará su tipo y devolverá una cadena que contenga el tipo de nombre o •ttpo desconocido• si no es uno de los tipos est4ndar, es decir, entero, doble, cadena, matriz u objeto.

Para utilizar aettype (>,le pasamos una variable cuyo tipo deseemos modificar y una cadena que contenga un nuevo tipo para dicha variable a partir de la lista anterior.

Nota

En este libro y en la documentación de php. ne t se hace referencia a un tipo de datos mixto. No existe como tal pero debido a la flexibilidad de PHP con respecto al control de tipos, muchas funciones pueden adoptar muchos tipos de datos como argumento. Los argumentos para los que se permiten varios tipos de muestran con esta modalidad de pseudo tipo de datos.

Podemos utilizarla de la siguiente forma:

```
$a• 56!
ecbo gettype($a).'cbr />';
••ttype(Sa, 'double•J;
ec:bo gettype(SaJ.'cbr />';
```

Cuando se llama a gettype (› por primera vez, $a es de tipo entero. Tras llamar a Ht type (›, el tipo se convierte en doble.

PHP también incorpora funciones para probar tipos. Cada l,lla de estas toma una vmable como argumento y devüelve früe ¡;-false:-w fuñaoiies soñ:

- is_array()
- is_double (), is_Uoat (), is_real () (Todas la misma función)
- is_long (), is_int (), is_integer () (Todas la misma función)
- is_string()
- is_object ()
- is_resource ()
- is_null ()
- is_scalar (>:Comprueba si la variable es escalar, es decir, de tipo entero, Booleano, cadena o flotante.
- is_numeric () :Comprueba si la variable es algún tipo de número o cadeftf nummca.
- is_callable(l:Comprueba si la variable es el nombre de una ñmción válida.

EL_HOLISTICO@HOTMAIL.COM - 2006

Probar el estado de las variables

PHP dispone de varias formas de probar el estado de una variable. La primera de estas formas es iaaet ¢>, que consta del siguiente prototipo:

```
bool 18..t(aixed var);
```

Esta función toma el nombre de una variable y devuelve true si existe y false en caso contrario. Tambifn puede pasar una lista de variables separadas por comas para que iseet ¢> devuelva true si se han establecido todas las variables.

Puede eliminar una variable utilizando unset ¢>• Éste es su prototipo:

```
void UDHt (aixed var) I
```

Esta función suprime la variable puada.

Por último, tenemos la función empty () . Esta función compi:ueba si existe una variable y si contiene un valor no vado o distinto a cero, y devuelve true o false según el caso. Su sintaxis es la siguiente:

```
boole•n eapty(aixed ver);
```

Vamos a examinar un ejemplo del uso de estu funciones.

Intente agregar el siguiente código a su secuencia de comandos temporalmente:

```
echo •iaaet(Stireqty):   •.iaaet(Stireqty).'cbr  />';
echo 'iaaetCSnothere),  '.iaaetCSnothere).  'd1r  />';
echo 'eaptyCStireqtyl:  '.eaptyCStireqty).'cbr   />'1
echo 'aaptyCSnothere):  •.eaptyCSnothere).'cbr  />'1
```

Actualice la p4gina para ver los resultad~

La variable $tireqty debería devolver 1 (true) de hset () con independencia del valor introducido o de si se introdujo algún va1or en el campo del formulario. Que sea empty ¢> o no depende del valor introducido.

La variable $nothere no existe, por lo que generar6 false desde isaet () y true desde empty ().

Estu funciones pueden resultar útiles para aseguramos de que el usuario rellenó los campos apropiados del formulario.

Reinterpretar variables

Puede lograr el equivalente de convertir una variable llamando a una función. En este sentido, existen tres funciones dtiles:

```
int intval(aixed ver);
rn—t doubhval (aixed varl I
atriag atrval|aixed varl;
```

Cada una de &tas acepta una variable como entrada y devuelve el valor de la variable convertida en el tipo apropiado. La función int val ¢> tambifn nos permite especificar la base de conversión para convertir una v~ble en una cadena, de esta forma podemos convertir cadenas hexadecimales en enteros.

Implementar estructuras de control

Las estructuru de control de un lenguaje permiten controlar el Oujo de la ejecución de un programa o secuencia de comandos. Las estructuru de control se pueden agrupar en estructuras condicionales (o de bifurcación) y en estructuras de repetición, o bucles.

En las siguientes secciones se examinarán las implementaciones específicas de cada una de ellas en PHP.

Tomar decisiones con estructuras condicionales

~i deseamos responder lógicamente a las entradas de nuestros us~os, nuestro código debe ser capaz de tomar decisiones. Fsta función recae sobre !Is estructuras condicionales.

Instrucciones if

Podemos utilizar una instrucción Ú para tomar.1uUl- decisión. Debemos darle una condición a la instrucción i f para que la utilice. Si la condición fuera true se ejecutará el siguiente bloque de código. Las condiciones de las instrucciones 'i f deben ir incluidas entre paréntesis.

Por ejemplo, si mandamos un pedido en el que no se incluyan neum6ticos latas de aceite ni bujías en el sitio de Bob, probablemente se debeni a que se ha p.;lsado accidentalmente el botón Submit Order. En lugar de indicamos "Order processed" la p4gina podría devolver un mensaje mucho ms útil.

Cuando el visitante realiza un pedido sin ningdn artículo, podríamos indic6rselo. Para ello podemos utilizar la siguiente instrucción if:

```
ifC Stotalqty •• o I
   ac:ho 'You did not order anythi119 oa ti.  previoua page|cbr  />';
```

La ~ción que estamos utilizando es $totalqty ... o. Recuerde que el operador iguales(••) se comporta de manera distinta al operador de asignación (•).

La condición $totalqty •• 0 ser6 true por lo que $totalqty es igual a cero. Si $totalqty no es igual a cero, la condición será false. Cuando la condición sea true, la instrucción echo se ejecutar6.

EL_HOLISTICO@HOTMAIL.COM - 2006

Bloques de código

A menudo necesitaremos ejecutar m'5 de una instruc:ción dentro de una secuen-c:ia condicional como i f. No es necesario colocar una nueva instrucción i f para cada una de ellas. En su lugar, podemos agrupar un número de instrucciones en un bloque. Para declarar un bloque, encimelo entre llaves:

```
if( $totalqty •• o I
(
   echo 'cfont color•re4>';
   echo 'You 414 not order aaytbing oa tbe pre•ioua pa9elcbr />';
   echo •c/f0Slt>':                                    •
```

Las tres JJneu de código encerrado entre llaves forman ahora un bloque de código. Si la condición es true, se ejecutarán las tres líneas. Si la condición es fal•e, se ignorarán las tres líneas.

Nota

Como se mencionó anteriormente, en PHP no es Importante la disposición del código. Sin embargo, conviene sangrarlo por cuestiones de legibilidad. Las sangrfas nos permiten distinguir de un vistazo qué lineas se ejecutarán sólo si se cumpten una condición, qué Instrucciones se agrupan en bloques y qué Instrucciones forman parte de bucles o funciones. En los ejemplos anteriores, se ha sangrado la instrucción que depende de la instrucción i f y las instruc-
~~s_q~~ ——n ~l ~c,qu l!·

Instrucciones else

Con frecuencia no sólo querrá decidir si desea que se ejecute una acción sino seleccionar una entre un conjunto deellas.

Las instiucáones e he permiten establecer la adopción de una acción alternativa cuando la condición de una instrucción i f resulte fal•e. Queremosavisar a los clientes de Bobsi envían una pedido sin ningún artículo. Por otra parte, si realizan un pedido, en lugar de una advertencia, queremos mostrarles lo que han pedido.

Si reorganizunOS nuestro código y agregamos una instrucción elae, podemos mostrar un avdO o un resumen de los artículos solicitados.

```
if( $totalqty •• 0 I
(
   echo 'You iu.4 -t order aa.ytllJ.n9 oa tbe previou• pagelcbr />';
}
•]••
(
```



```
   echo $t1reqty.' tire•cbr />';
   echo $oUqty.' botth• of oilcbr />';
   echo $•parkqty.• •park plu98cbr />';
```

Podemos desarrollar procesos lógicos mú complejos anidando instrucciones if. En el siguiente código, nosólo se mostraral el resumen si la condición $totalqty •• 0 resulta ser cierta, sino que ademú cada línea del resumen sólo se mostrará si se cumple su propia condición.

```
ifl $totalqty •• 0I
(
   echo 'You did not ord•r aaytbing on tbe previou• pa9elcbr />';
}
el••
(
   if ( $tireqty>0 I
      echo $tireqty.• tire•cbr />';
   1f ( $oilqty>0 I
      ecba $oilqty.• bottl•• of oilcbr />'1
   if 1 $eparkqty>0 I
      ecba $eparkqty.• •park plug•cbr />';
```

Instrucciones elseif

Para muchu de las decisiones que tomamos suele haber má de dos opciones. Podemos crear una secuencia de variu opciones utilizando la instrucción elHif. Esta instrucción es una combinación de elae e if. Al suministrar una secuencia de ~~~~~e!_prop-ama P-~~~compro.bar~~~:l!'ll. ~e ellas.~~~~~en~L una que sea true.

Bob ofrece un descuento por grandes pedidos de neumáticos. La oferta se articula de la siguiente forma:

- Menos de 10 neumáticos, sin descuento
- De 10 a 49, 5% de descuento
- De 50 a 99, l()'ll, de descuento
- 100 o mú de 100,15% de descuento

Podemos crear código para calcular el descuento utilizando condiciones e instrucciones if y elseif. Necesitamosutilizar el operador ANO (u) para combinar las dos instruccionesen una.

```
if ( $t1nqty e 10 )
   $4i•count • 0;
eleeifl $tireqty >• 10 •• $tireqty e• 49 1
   $4bcouat • S,
el••ifl $t1reqty >• 50 •• $tireqty e• 99)
   $clbcount • 10;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
elaeifl  Stireqcy  >• 100  1
   Sdiacount  •  151
```

Fíjese en que da lo mismo escribir el—if o elae if, ya que ambasson correctas.

Si va a escn"bir un conjunto de instrucciones elaeif en cascada, debería ser consciente de que sólo se ejecutad uno dé los bloques o instrucciones. En este ejemplo no importa porque todas las condiciones son mutuam~te excluyentes, sólo una puede ser verdadera a la vez. Si escribiéramos las condiciones de forma que más de una pudiera ser verdadera, sólo se ejecutaría el bloque o instrucción situada a continuación de la primera condición cierta.

Instrucciones switch

La instrucción awitch funciona de una forma similar a la instrucción if, pero permite que la condiáón tome más de dos valores. Enuna instrucción i f, la condición puede ser true o falae. En una instrucción awitcb,. la condición puede tomar cualquier número de valores diferentes, siempre y cuando se eval\\e en un tipo \\nico (entero, cadena o doble). Es necesario incluir una instrucción case para cada valor al que desee reaccionar y, opcionalmente, una instrucción case prede-~enninada para procesar aquellos valores pana los que no se hayan incluido una instrucción case especifica. Bob quiere saber qué tipo de publicidad atrae visitantes a su sitio. Para ello podemos agregar una pregunta a nuestro formulario de pedidos. Inserte el siguiente código HTML dentro del formulario de pedidos y el formulario presentad un aspecto parecido al ilustrado en la figura 1.6.

```
cCr>
   ctd»Row did you fiad Bob'ac/td>
   ctd»caelect,aa ..••find•»
            coption value • 'a•»I'•  a regular cuatomer
            coption value • °b°»TV  advertieiag
            coption value • 'c•»Pboae directory
            coption value • °d°»110rd  of aoutb
         c/Hlect>
   c/td»
c/tr>
```

Este código de H1ML agrega una nueva variable de formulario cuyo valor seré •a•, "b", •c• o •d•. Podríamos procesar esta nueva variable con una serie de instrucciones if y elae if como la siguiente:

```
if(SfiDd •• 'a')
   edlo  'cp»Regular cuatomer.c/p»•;
elaeif(Sfilld •• 'b')
   edlo 'cp»Cuatomer  referred by  TV aclvert.c/p»'1
el..if(Sfilld •• 'e•)
   ec:llo 'cp»C:U.tomer  referred by  pbcnle directory.c/p>'1
elaeif(Sfind •• 'd')
   edlo  •cp»CU•t-r referred by  word  of  —tb.c/p»•;
```

Figura 1.1. El fonnul8rlo de pedidos pregunlll a los visitantes cómo encontraron el litio Bob'I Auto Parta.

Como alternativa podriamos utilizar una instrucción switch:.

```
avitch (Sfilld)
(
   e•••  •a•  :
      eclla 'cp»Regular  cuatoaer.c/p>':
      break;
   ca••  •b•
      eclla 'cp»CUat — r  referred by  TV advere. c/p>':
      breaic;
   e•••  •e•  :
      eella 'cp»CU•t — r  referred by phone directory.c/p>'1
      break:
   e•••  •d•
      eclla 'cp»CUatomer  referred by  vord of moutb.c/p>';
      break;
   default,
      ecbo 'cp»lfe do not luaow hov tbia cuat — r found ua. c/p>':
      break;
```

(Apreciar6 que en ambos ejemplos se supone que ha extmfdo $f ind de la matriz $ POST.)

–la instrucción switch se comporta de una forma ligeramente diferente a una instrucción if o elaeif. Una instrucción if afecta sólo a una instrucción a menos que se utilicen llaves de manera deliberada para crear bloques de ones. Una instrucción awitch se comporta de forma contraria.

Cuando se activa un caso en una instrucción swi tcb. PHP ejecutad las instrucciones hasta que alcance una instrucción break. Sin una instnlcción break, la instrucción awi tch ejecutaría todo el código situado detrú del caso que resultara

EL_HOLISTICO@HOTMAIL.COM - 2006

ser cierto. Al alcanzar la instrucción break, se ejecutar, la lfnea de código situada tras la instrucción switch.

Comparar condiciones diferentes

Si no est, familiarizado con estas instrucciones, es probable que se esti pregun· tando cu¸l es la mejor.

Sin embargo, no podemos dar una respuesta exacta a esta pregunta. No hay nada que pueda hacer con una o varias instrucciones elae, elseif o awitch que no pueda hacene con un conjunto de instrucciones if. Qebería intentar utilizar aquella opción que le resulte más sencilla de leer. Con el tiempo ir, aprendiendo a seleccionar una u otra.

Iteración: repetir acciones

Una de las cosas para las que los ordenadores han demostrado siempre ser buenos es para alltOmatizar tareas repetitivas. Si hay algo que necesite hacer de la misma forma una serie de veces, puede utilizar un bucle para repetir partes de un programa. Bob quiere una tabla que muestre el coste de envío en función de la distancia a la que se está enviando el paquete. El coste se puede calcular con una se,:1pllafórmula. Queremos que la tabla de costes de envío se parezca a la ilustrada en~· figura 1.7.

Flguni 1.7. Estll tabla muesn el coste de envio en función de la distancia.

El listado 1.2 incluye el código HIML utilizado para mostrar esta tabla. Como obse~ar¸, el código es largo *y* repetitivo.

Listado 1.2. freight.html. Tsbla de costes de envio de Bob.


```
<table border••o•  cellpacldiJlg•"l">
ctr>
   ctd bgcolor••tcccc:cc•   align••center">Diet&llleec/td>
   ctd bgcolor•"ICCCCCC"   align••ceater•>Coetc/td>
c/tr>
ctr>
   ctd align•"right">50c/td>
   ctd align••right">Sc/td>
c/tr>
ctr>
   ctd align••right">100</td>
   ctd aliga••right">l0</td>
c/tr>
ctr>
   ctd aliga••right">lSOc/td>
   ctd aliga••right">lSc/td>
c/tr>
ctr>
   ctd align•"right">200c/td>
   ctd aliga•"rigbt">20c/td>
c/tr>
ctr>
   ctd align••right">250c/td>
   ctd aliga••rigbt">25</td>
c/tr>
e/table>
c/'body>
c/htmb
```

Convendría delegar en un ordenador barato e incansable la tarea de escribiJ: este código de HTML en lugar de recurrir a un humano que se aburriría con facilidad (y· qμe qμe pagad, Las ones c:Je !,μele ha~ que .ll:fe_ejecutuma.. instrucción o un bloque de manera repetida.

Bucleswhile

El tipo de bucle más sencillo en PHP es el bucle while. Como en el caso de las instrucciones if, se basa en una condición. La diferencia entre un bucle while *y* una instrucción i f es que 6sta ejecuta el siguiente bloque de código si la condición resulta ser true. Un bucle while ejecuta el bloque repetidamente mientras la condición sea true. Los bucles while se utilizan cuando no se sabe cumtas iteraciones resultará necesarias para que la condición resulte cierta. Si el número de iteraciones debe ser fijo, considere la posibilidad de utilizar el bucle for.

La estructura búica de un bucle whil e es la siguiente:

```
wbile(  coadici6a)   exprHidar
```

El siguiente bucle mostrar, los números del 1 al S.

```
$a—• 1,
while  ($nu• e• 5 l·
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
   ecbo Snwa.•cbr /»•;
   $AUll•+;
```

Al principio de cada operación, se prueba la condición. Si es falsa, el bloque no se ejecuta y el bucle finaliza. A continuación, se ejecutar4 la instrucción situada por detrás del bucle. Podemos utilizar un bucle whi le para realizar algo mú útil como mostrar la tabla de costes de envio repetitivos de la figura 1.7. El listado 1.3 utiliza un bucle while para generar la tabla de gastos de envio.

Listado 1.3. freight.php. Generación de la tabla de gutoa de envio de Bob en PHP.

```
c!:,ody»
ctable border•"O" cellpadding•°l">
ct.r>
   ctd bgcolor•"ICCCCCC• align••cancer•»Diatancec/td>
   ctd i,gcolor•"ICCCCCc• align••cencar•»Coatc/td>
c/tr>
e?
Sdiatanca • S0,
vbile ISdiatance •• 250
(
   acbo •ctr»\n ctd align•.•rigbt">Sdietancac/td»\n•1
   acbo • ctd align••rigbt•»•. Sdiatanc• / 10 .•c/td»\nc/tr»\n•;
   Sdiatanc• +• S0,

?»
c/tüla>
c/body»
c/htal»..  ------- --- -
```

Para que el código HTML generado por nuestra secuencia de comandos resulte legible, debe incluir nuevo lfneu y espacios. Como se indicó anteriormente, los navegadores ignoran estos elementos pero resultan importantes para facilitar la lectura de los humanos. Con frecuencia necesitar6 examinar el código HMTL si el resultado obtenido no es el esperado. En el listado 1.3 verá que algunas cadenu incluyen los caracteres \n. Si se incluye dentro de una cadena encerrada entre comillas dobles, representa un car4cter de nueva linea.

Bucles for y foreach

La forma en que utilizamos los bucles whi le anteriormente resulta muv común. En primer lugar establecimos un contador. Antes de cada interacción, probamos el contador en una condición. Al final de cada iteración, modificamos el contador.

Pod~mos escribir este estilo de bucle de forma m6s compacta utilizando un bucle for. La estructura búica de los contadores for es la siguiente:

```
forC upreai6D1; conclici6nl exprHieDZ)
   upreai6nl1
```


- La expreai6nl se ejecuta una vez al principio. En este parúnetn, se suele establecer el valor inicial de un contador.
- La condici6n se prueba antes de cada iteración. Si la expresión devuelve false, la iteración se detendr4. En este parámetro se suele probar el contador con respecto a un limite.
- La expreei6n2 se ejecuta al final de cada iteración. En este parámetro se suele ajustar el valor del contador.
- La expreai6n3 se ejecuta una vez por iteración. Esta expresión suele ser un bloque de código y contendrá el grueso del código de bucle.

Podemos rescribir el ejemplo del bucle whi le del listado 1.3 como un bucle for. El código PHP se convertir6 en

```
c?pbp
forCSdiatanca • 50; $diatance e• 250! $diatance +• 501
(
   echo •ctr»\n ctd align•'rigbt'>Sdiatancac/td»\n•1
   echo•  ctd align•'rigbt'>".  $diatanc:• / 10 .•c/td»\nc/tr»\n•,
7>
```

Tanto la versión while como la versión for funcionan de man$ idéntica. El bucle for resulta un poco m6s compacto y ahorra dos lineas.

Ambos tipos de bucles son equivalentes. Ninguno de los dos es1ro peor que el otro. En una situación dada, puede utilizar el que le resulte m6s •_"tivo.

Como comentario al margen, hay que decir que se puede comb· una variable de tipo-variable con un bucle for para procesar una iteración a:travá de una serie de campos de formulario repetitivos. Si, por ejemplo, tiene campos de formulario con nombres como namel, name2, name3, etc., puede procesarlos de la siguiente forma:

```
for 151•11 Si e• $nuamaa••1 Si++I
(
   t•IIP• •na•Si•;
   cbo SSt•1l(|).'cbr />';//o  cualquier proc••••iento que de••• realisar
```

Al crear dinúnicamente los nombres de las variables, podemos acceder a cada campo uno por uno.

Adem6s del bucle for existe el bucle foreach, diseñado específicamente para su uso con matrices. En un capitulo posterior comentaremos cómo utilizarlo.

El tipo final de bucle que mencionaremos se comporta de modo lip,ramente diferente. La estructura general de una instrucción do••• while es la siguien~:

EL_HOLISTICO@HOTMAIL.COM - 2006

```
do
   expre<l6n!
vbile( condici6n !;
```

Un bucle do .. while se diferencia de un bucle while en que la condición se prueba al final. Por lo tanto en un bucle do .·. while, la instrucción o el bloque incluido en el bucle se ejecuta siempre una vez al menos. Incluso si tomamos un ejemplo en el que la condición serli false al principio y nunca puede ser true, el bucle se ejecutarli una vez antes de comprobar la condición y el final.

```
$Diia • 100;
do
{
   echo $nua.•cbr />';
}
vhile (Snua e 1 );
```

Salir de una estructura de control o una secuencia de comandos

. Si desea detener la ejecución de un fragmento de código, existen tres opciones que dependen del efecto que estt§ persiguiendo.

Si desea detener la ejecución de un bucle, puede utilizar la instrucción break como se comentó anteriormente en la sección sobre switch. Si utiliza esta instrucdón en un bucle, la ejecución de la secuencia de comandos continuad en la lfnea situada bU el bucle. SLdesea.sal~ta..la.siguiente .iterac:i6n...de bucle.. puede utilizar la instrucción cont inue. Si desea terminar de ejecutar la secuencia de comandos PHP entera, puede utilizar la instrucción exi t. Esta opción resulta útil al realizar tareas de comprobación de errores. Por ejemplo, podemos modificar nuestro ejemplo anterior de la siguiente forma:

```
if( Stotalqty •• 0!
{
   echo 'YOu did not order anything on the previoue pege!cbr />'i
   eaiti
```

La llamada a exi t impide que PHP ejecute el resto de la secuencia de comandos.

Utiliza una sintaxis alternativa de estructuras de control

Para todas las estructuras de control que hemos visto hasta el momento existe un sintaxis alternativa. Consiste en sustituir la llave de apertura (() por dos puntos

(:) y la llave de cierre con una nueva palabra clave como endif, endswitch. endwhile, endfor o endforeacb. en función de la estructura de control utilizada. No exiate una sintaxis alternativa pua los bucles do ··· while.

Por ejemplo, al código

```
if( $totalqty •• 0)
(
   echo 'You did not order anytbi119 on tlle previoue page I cbr />•!
   exit;
```

se le podría aplicar esta sintaxis alternativa por medio de las palabras clave if y endif:

```
ifl $totelqty •• 0l,
   ecllo 'You did not order anytbi.ng on the previoue pagelcbr />';
   exit,
endifl
```

Utilizar declare

Otra estructura de control de PHP es la estru~tura declare, que no se utiliza con tanta frecuencia como otras estructuras de programación. El formato general que emplea es el siguiente:

```
declare (directiva)
{
   1/-bloqua ..
]
```

Esta estructura se utiliza para establecer directivas de ejecución para bloques de código, es decir, para determinar la forma de ejecutar el código. En la actualidad, sólo se ha implementado una directiva de ejecución, ticks. Para establecerla debe añadir la directiva ticks•n. Le permite ejecutar una determinada función cada n líneas de código dentro del correspondiente bloque, lo que generalmente se utiliza para crear perfiles y para depuradón.

Sólo mencionamos la estructura de control declare a tftulo informativo y en capítulos posteriores incluiremos algunos ejemplos de uso de las funciones tick.

Siguiente paso: guardar el pedido del cliente

Ahora ya sabe cómo recibir y manipular pedidos de un cliente. En el ,siguiente capftalo examinaremos cómo almacenar el pedido para poder recuperarlo y servirlo.

EL_HOLISTICO@HOTMAIL.COM - 2006

2

Almacenar y recuperar datos

Ahora que ya sabemos cómo acceder y manipular datos introducidos en un formulario HTML, podemos pasar a examinar varias formas de almaftMr dicha información para su uso posterior. En la mayor parte de los casos/1nduido el ejemplo examinado en el capftulo anterior, querrá almacenar los datos y cargarlos en un momento posterior. En nuestro-caso-necesitamos escribir pedidos de cliente para almacenarlos y poder servirlos después.

En este capítulo vamos a explicar cómo podemos escribir el pedido del cliente del ejemplo anterior en un archivo y cómo volverlo a leer. También veremos por qué esta solución no es siempre la mejor. Sí tenemos un gran número de pedidos, deberiamos utilizar un sistema de administración de base de datos como MySQL

Entre los temas clave que trataremos en este capítulo se incluyen los siguientes:

- Guardar datos para su uso posterior
- Abrir un archivo
- Crear y escribir en un archivo
- Cerrar un archivo
- Leer un archivo
- Bloquear un archivo
- Eliminar archivos
- Otras funciones útiles con archivos
- Una opción mejor: los sistemas de administración de bases de datos

EL_HOLISTICO@HOTMAIL.COM - 2006

Guardar datos para su lectura posterior

Básicamente existen dos formas de almacenar datos: en archivos planos o en una base de datos. Los archivos planos pueden tener múltiples formatos pero, en general, cuando hacemos referencia a un archivo plano, nos estamos refiriendo a un archivo simple de texto. En este ejemplo, vamos a esaibir los pedidos de los clientes en un archivo de texto, uno en cada línea. Este método es muy sencillo, pero resulta a la vez muy restrictivo, como veremos en una sección posterior. Si está tratando con información de un volumen significativo es probable que prefiera utilizar una base de datos. Sin embargo, los archivos planos se utilizan en determinadas ocasiones y hay situaciones en las que necesitan saber cómo hacerlo. Las operaciones de lectura y esaitura d!! archivos en PHP se realizan prácticamente de la misma forma que en C. Si ha programado en C o ha desurollado secuencias de comandos de núcleo para UNIX, la mecánica le resullar.6 bastante familiar.

Almacenar y recuperar los pedidos de Bob

En este capitulo, utilizaremos una versión modificada del formulario de pedido visto en el último capítulo. Comenzaremos por este formulario y por el código de PHP que escribimos para procesar los datos del pedido.

Hemos modificado el formulario para incluir un m~o rápido para obtener la dirección de envío del cliente. En la figura 2.1 se ilustra su aspecto.

El campo del formulario en el que se recoge la dirección de envío del cliente se denomina address. Este campo nos da una variable a la que podemos acceder como $ RBQUEST ('address ' J, $ POST ('address ' J o $ GBT ('address' l, en función del ntodo de envío utilizado. Escribiremos cada pedido que entre en el mismo archivo. Seguidamente, construiremos una interfaz Web para permitir que la plantilla de Bob pueda ver los pedidos que entran.

Procesar archivos

La operación de escribir datos en un archivo induye los siguientes pasos:

1. En primer lugar, abrir el archivo. Si el archivo no existiese, tendrfamos que aearlo.

2. Escribir los datos en el archivo.
3. Cerrar el archivo.

Figura 2.1. Esta versión del pedido obtiene la dlrecc:i6n de envio del diente.

La operación de lectura de los datos de W) archivo tambibl incluye de tres pasos:

1. Abrir el archivo. Si no se puede abrir (por ejemplo, si no existiese), tendríamos que reorganizar el proceso y salir con elegancia.
2.- l.eer los da'tos ae hrchivo.
3. Cerrar el archivo.

Cuando queremos leer los datos de un archivo, podemos determinar quf canti; d~ de~ent os del archivo leer cada vez. En 1U1a sección posterior examinaremos dtchas opaones. En primer lugar vamos a comenzar por abrir un archivo:

Abrir un archivo

Para abrir un archivo en PHP, se utiliza la función f open (). Al abrir un archivo tenemos que especificar para quf tenemos pensado utilizarlo. Es lo que se co~ como los modos de archivo.

Seleccionar modos de archivo

El sistema operativ~ del servidor necesita saber quf queremos haéer con el archivo que vamos a abrir. Necesitamos saber si otra secuencia de comandoe puede abrir

EL_HOLISTICO@HOTMAIL.COM - 2006

el archivo que ya tenemos abierto y determinar si el titular de la secuencia de comandos dispone de permiso para utilizarlo de esa forma. Búicamente, los modos de archivo ofrecen al sistema operativo un mecanismo para determinar la forma de procesar las peticiones de acceso procedentes de otru personas o secuencias de comandos, y un método para comprobar si dispone de acceso y permiso para utilizar el archivo. Al abrir un archivo dispone de tres opciones:

1. Puede abrir un archivo para leerlo, para escribir en él o para ambas acciones.
2. Si está escribiendo en un archivo, puede sobrescribir los contenidos existentes o adjuntar nuevos datos al final del archivo. También puede que le interese finalizar el programa de forma elegante en lugar de sobrescribir el archivo si éste ya existe.
3. Si está intentando escribir en un archivo sobre un sistema operativo que distinga entre archivos binarios y archivos de texto, puede que desee especificar esta circunstancia.

La función fopen () admite combinar estas tres opciones.

Utilizar fopen() para abrir un archivo

Supongamos que desea escribir el pedido de un cliente en el archivo de pedidos de Bob. Puede abrir el archivo para escribir el pedido con la siguiente secuencia:

```
$fp.    fopen(•$DOCUMIIIT_ROOT/ ·. /ordere/ordera.txt•,    •w•11
```

'Ar llamar a fopen ( ) ; se esperan-dos, tres o cuatro parámetros. Por ,;egla g~~ral se utilizarán dos, como se muestra en la linea de código.

El primero debería ser el archivo que queremos abrir. Puede especificar una ruta hasta este archivo como hemos hecho en el código anterior; nuestro archivo orders ·txt se encuentra dentro del directorio de pedidos. Hemos utilizado la variable incorporada $HT1'P_sERVER_VARS C' OOCtJMBNT_ROOT' I pero, como ocurre con los incómodos nombres completos de las variables de formulario, le hemos asignado un nombre más corto.

Esta variable apunta a la base del árbol de documentos de su servidor Web. Hemos utilizado • .. • para indicar "el directorio superior del directorio rafz del documento". Este directorio se encuentra fuera del úbol de directorios, por razones de seguridad. No queremos que este archivo resulte accesible desde la Web salvo a través de una interfaz que proporcionaremos nosotros. Esta ruta se conoce como ruta relativa ya que describe una posición en el sistema de archivos en relación al directorio del documento.

Como ocurre con los nombres cortos que asignamos a las variables de formulario, si no activamos el parámetro register_globals, necesitamos incluir la siguiente lín~ al principio de la secuencia de comandos:

```
sl)()CIJMSll'l'_ROOT • $HTTP_sn.vn_ vus l' DOCUNBIIT_ROO"r • l ;
```


para COfiar los contenidos de la variable de estilo largo a la variable de estil f°~to. ~l igual ~ue existen varias formas de acceder a los datos de formu~ aoo: ex1Sten vanas formas de acceder a las variables de servidor predefinidu En ==~ : e la configuración del servidor puede acceder a la raíz del documento a

- $DOctJMBNT_ROOT
- $_SBRVBR['DOctJMENT_ROOT')
- $HT1'P_SBRVBR_VARS ['DOCUMENT_ROOT']

~omo en el caso d~ los datos de formulario, el primer estilo es el preferido.

am~1én se espec1fica una ruta absoluta al archivo. &ta es la ruta desde el :~::nora~(/ en los sis~s UNIX y, por regla general, e:\ en Windows). En serv1 or.UNJ?C• seria /~~/book/ orders. El pro.,_ de utilizar este mT~° es que si aloiamos el sitio en el servidor de un tercero, la ruta absoluta po na~~nar. nosotros nos tocó aprenderlo de la forma difícil ya que tuvimos que m car ~ rutas absolutas de una gran cantidad de secuencias de comandos cu~o los a~is~adores del sistema decidieron cambiar la estructura de diiectonos sm previo av1So.

S~o se «:5pecifica~ ruta, el archivo se creará o se buscará en el m¡;mo directorio que propia ~enaa de comandos. &ta será diferente si está ejeostando PHP a trav~de algún tipo de contenedor CGI y ello dependerá de la coafiguración del sem or.

En~ entorno UNIX, las barras de los directorios son las barras IMándar (/). Si ~ ~tiliz = do~aplataforma Windows, puede usar éstas o las barrdlnvertidas (\). ~a . tem Cas las timasrdebe-marcarlas-comocaracteres deescapepara~ue fopen-(). derlas correctamente. Para marcar un carácter como carácter especial ta con agregar !)tra barra invertida delante, como se muestra a continuación:

```
$fp • fopenl"S~IDIT_ROO'r\\ •• \\orden\\order•. txt•, •w•) 1
```

ódLas b~ in':ertid~ no se.suelen utilizar para indicar rutas en PHP porque el código só o fun~onaria. en.Wmdows. El uso de la barra estándar permite mover e go entre eqwpos Wmdows y UNIX sin variaciones.

1l ~do p~etro de fopen () es el modo de archivo, que debería ser una ca ena ta es~~~ qué hacer con el archivo. En este caso vamos a pasar , w, a fop-:n !>, que significa abnr el archivo para escritura. En la tabla 2.1 se descr1ben los d1Stintos modos de archivo.

Tabla 2.1. Resumen de los modos de archivo de topen().

[illegible]	[illegible] modo	[illegible]
r	Lectura	la parte Inicial del archivo.

EL_HOLISTICO@HOTMAIL.COM - 2006

Modo	Nombre del modo	Significado
r+	Lectura	Abre el archivo para lectura y e~ra, empezando por la parte inicial del archivo.
w	Escritura	Abre el archivo para escritura, empezando por la parte inic:lal del archivo. Si el archivo ya existiese, elimina sus contenidos. Si no exis-ttese, intenta crearlo.
w+	Escritura	Abre el archivo para escritura y lectura. Si el archivo ya existiese': elimina sus contenidos. Si no existiese, intenta crearlo.
x	Escritura con advertencia	Abre el archivo para escritura, empeza~ por la parte Inicial del archivo. Si 6ste ya ex11tlese, no se abre, fopen <I devuelve false Y PHP genera una advertencia.
x+	Escritura con advertencia	Abre el archivo para escritura y lectura, empezando desde la parte Inicial del mismo. Si ya existlese,noseabre,fopen<I dewelvefalae y PHP genera una advertencia.
a	Adjunción	Abre el archivo para adjuntar (escribir) únicamente empezando por la parte final de los conte~idoS existentes. Sl no existiese, intenta crearlo.
a+	---~j~~i~-	Abre el archivo para adjuntar (escribir) Y I~r, empezando por.la parte final.de.io. con~n~ existentes. Si no existiese, intenta crearlo.
b	Binario	se utiliza en combinación con uno de los otros modos. Puede utilizarlo si su sistema de arch~vos distingue entre archivos binarios Y archivos de texto. Los sistemas WIndows hacen esta diferencia al contrario que los sistemas UNIX. Los programadores de PHP le ~ miendan que siempre utilice esta opción ~ obtener la máxima portabilidad. Es el modo predeterminado.
t	Texto	se utiliza en combinación con uno de los otros modos. Es una opción exclusiva de WIndOWI. Noes recomendable a menos que haya adapllt-do su código para que funcione con la oPción b.

El modo de archivo para nuestro ejemplo depende de cómo se utilice el sistema. En nuestro caso, hemos utilizado 'v" que sól~ permitebal~= :' ~:~. ~ archivo. Cada vez que se tome un nuevo pedido, se so rescn

opción no parece muy adecuada por lo que haremos mejor en seleccionar el modo de adjunción (y el modo binario, como hemos recomendado).

```
$fp • fopen("$DO<:UMBIIT_ROOT/.' /ordera/ordera.txt•,  'ab'I;
```

La función fopen () ~ta de un tercer parámetro opcional. Puede utilizarlo si desea buscar el parámetro inc:l ude_path (establecido en la configuración de PHP; a este respecto consulte el apéndice A) de un archivo. Si desea activar esta opción, asigne I al parámetro. De esta forma no tendrá que suministrar un nombi:eo una ruta de directorio:

```
$fp • fopen(•ordera.txt•,  'ab'• true) I
```

El cuarto parámetro también es opcional. La función fopen (I permite que los nombres de archivo tengan como prefijo un protocolo (como por ejemplo ht tp : / /) y que se abran en una ubicación remota. Algunos protocolos permiten un parámetro adicional. Analizaremos este uso de fopen (I en una sección posterior.

Si fopen (I se abn! satisfactoriamente, se devolverá un puntero al archivo que deberla guardarse en una variable, en este caso $fp. Esta variable se utiliza para acceder al archivo cuando se desee leer o escribir en él.

Abrir archivos a través de FTP o HTTP

Además de abrir archivos locales para su lectura o escritura, puede abrir archivos a través de FTP, HTIP y otros protocolos utilizando fopen (l. Puede anular esta posibilidad si desactiva la directivaallow_url_fopen delarchivophp-ini. Si tiene problemas para abrir archivos zemotos pounedio de.fopen.(), compAiebe su archivo php. ini.

Si el nombre del archivo utilizado comienza por ftp : //,se abrirá una conexión FTP en modo pasivo al servidor especificado y se devolverá un puntero al inicio del archivo. Si el nombi:e del archivo utilizado comienza por http: //, se abrirá una conexión HTIP al servidor especificado y se devolverá un puntero a la respuesta. Al utilizar el modo HITP, debe especificar las barras situadas al final de los nombres de directorios, como se muestra a continuadón:

```
bttp,//WWt1.aerver.coa/
```

no

```
http,//Wtltl.aerver.coa
```

Cuando se especifica la última forma de dirección (sin la barra), el servidor Web suele utilizar una n!direcáón HITP para enviarle a la primen dirección (con barra). Pruebe en su navegador.

En las versiones de PHP anteriores a la 4.0.5, la fundón fopen () no admitia redireccimes HITP, por lo que debe especificar los URL a los que luÚ:en referencia los directorios con una barra final.

EL_HOLISTICO@HOTMAIL.COM - 2006

A artir de la versión 4.3.0 tambiin puede abrir archivos sobre SSL sie°:'~re Y cuan/ose haya compilado o habilitado la compati~ilidad para OpenSSL y utilice la secuencia ht tp•: / / delante~~ ::::o~ no discriminan entre mayúscul~u=~~os Nta y el nombre del archivo puede que sf lo hagan.

Problemas al abrir el archivo

Un error que suele producirse de manelrahabitual par~ e[l que ~o s e disponga de permiso de ectura o escnuu tivos similares a Unix aunque también puede producirse habitual en SJStemasoperará similar al ,...,. se ilustra en la figura 2.2. en Windows.) PHP gehera un av

Figura z.2. PHP le avisa especlflcllmente cuando no puede abrir un archivo.

5 :be este error, debe asegurarse de que el usuario utilizado para eje~tar la de comandos dispone de permiso para acceder al archivo q~e está mtensecue~. fun .6n de cómo se configure el servidor, la secuencia de comantando uedtila.zin , ~e como el usuario de servidor Web o como el propietario del dos pu e que se eiecu manci directorio en el que se incluy~ la secuelanaade co d os~os se ejecuta como el En la mayor parte de los sJStemas, secuenaa e co usuario de servidor Web. Si su secuencia de comandos estuVJera en un.SJS~a t.n,i,X dentro del directorio -/publ.ic_h~m~/ chapter2 /, se cpedlre ur:::n~ sobre el que todo el mundo podría escnbir y almacenar el siguiente secuencia:.

```
üdir -/order•
cbllod 777 -/order•
```


Tenga en cuenta que los directorios y los archivos sobre los que puede escribir todo el mundo son peligrosos. No deberla utilizar directorios suscepb"bles ele escritura que resulten accesibles desde la Web. Por esta razón, nuestro directorioorders se encuentra dos subdirectorios mú atrú, por encima del directorio public html. ~ tema de la seguridad se comentará~ mú profundidad en un capítulo posterior.

Uno de los elementos que suele dar errores al abrir un archivo es el uso de permisos incorrectos, aunque no es el único problema. Si el archivo no se puede abrir, necesitará saberlo para no intentar leer o escribir datos en fl.

Si la llamada a fopen (1 falla, la función devolverá false. Puede tratar el error de forma que resulte mú significativo sustituyendo el mensaje de PHP por otro.

```
• Sfp • fopen("SDOCUMBJIT_II00T/ •• /order•/order•.txc•,  'ab');
u  (!$fp)
(
   ec:bo 'cp>c8trong>  Your ord8r could not be proc••••d at tlli•  ti...
         ' 'Plea•• try again later.c/8trong>c/p>c/body>c/htal>';
   exit;
```

El símbolo• situado delante de la llamada a fopen (1 indica a PHP que suprima todos los errores producidos por la llamada de función. Por regla general, conviene saber cuándo ocurre algún error, pero en este caso vamos a tratar el problema en otro lugar.

Esta linea también se puede escribir de la siguiente forma:

```
Sfp • efopen(•$D0c:tJIIBIIT_ROOT/ .. /order•/order•.txt•,  •a•);
```

Sin embargo, con esta secuencia resulta menos obvio que se esti utilizando el operador de supresión de errores. En un capítulo posterior se tratará el tema de la generación de informes de error de manera mú detallada.

La instrucdón if compNeba la variable $fp para determinar si se ha devuelto un puntero válido de archivo desde la llamada fopen. En caso negativo, imprime un mensaje de error y concluye la ejecudón de la secuencia de comandos. Como la página termina aquf, se han utilizado etiquetas HTML de derre para que el código HTML resulte correcto. En la figura 2.3 se ilustra el resultado de este enfoque.

Escribir en un archivo

La operación de escribir en un archivo en PHP resulta relativamente sencilla. Puede utilizar la función fwri te (1 (escribir archivo) o la función fputs (1 (cadena de archivo); fputs (1 es un alias de fwri te (1. En el siguiente ejemplo llamamos a la función fvri te (1:

```
fvriteCSfp, $output•tring);
```

Esta funci6nindica a PHP que escriba la cadena almacenada en $outputstring en el archivo al que apunta $fp.

EL_HOLISTICO@HOTMAIL.COM - 2006

Una nueva alternativa a fwrite () es la función file_put_~ontenta ()'cuyo prototipo es el siguiente:

```
iat file_put_eoateat• ( •triag aOllbre de erebivo,
        atring detoe
        !, iat iadieedore•
        !, reeouree eontextol 1 1
```

Es; ..... ~ct'.vo escribe la cadena incluida en datos en un archivo con el nombre de nom . d PHP 5 e·uivale a file get contenta (l' que ana sin necesidad de invocar fopen (l ni fcloae (l. Es una... • nuevas funaones e y e. ionales *indicadora-y contexto* se°suelen utilizar al mos en breve. Los parAmetros pe ed° d HT1'P o FI'P, como veremos en un capf• escribir en archivos remotos por m 10 e tulo posterior.

f111ura 2.3. El uso de mensajes da enor _propiOS en lugar de los de PHP pu~ mejorar la comprensiónde los errores.

Parámetros de fwriteO

La función fwrite (1 toma tres parámetros aunque el tercero es opaonal. Su sintaxis es la siguiente:

```
iat fvrite ( iat puntero, atriag ce d ene ℓ, lat longitud))
```

lon "taul es el número máximo de bytes que escribir. 51 se El te~ parámetro, ~ , "b; á la cadena en el archivo al que apunte el incluye este parámetro, fwri te '() escn ir d ha scrito la longitud parámetro*puntrro* hasta que alcance el fina~ de la ca :- ob~-: la longitud de la de bytñ, depen~iendo de q':1é apar;zca r,ri:;e;;;~oS:o°se indica a continuación: cadena por medio de la función str en ,

```
fvrite($fp,     $output•triag,     atrlea($outputatriag))¡
```


Puede que le interese utilizar este tercer parámetro para escribir en modo binario ya que permite evitar problemas de compatibilidad entre plataformas.

Formatos de archivo

Al aear un archivo de datos como el utilizado en el ejemplo, la decisión sobre el formato en el que almacenar los datos será nuestra. (Sin embargo, si tiene previsto utilizar el archivo de datos en otra aplicación, puede que necesite seguir las reglas de dicha aplicación.) Vamos a crear una cadena que represente un registro en nuestro archivo d~ datos. Para ello, podemos utilizar el siguiente código:

```
soutputatrlag • $date.•\t".$tireqty.•     tires \t•.$oilqty.•     oil\t•
        .$aperkqt7.•     spark pluga\t\$".$total
        .•\t•.     Saddreaa.•\a•¡
```

En nuestro sencillo ejemplo, estamos almacenando cada registro de pedido en una línea separada del archivo. Hemos optado por escribir un registro por línea porque de esta forma obtenemos un separador de registros sencillo en forma del carácter de nueva lfnea. Como las nuevas líneas son invisibles, las representamos con la secuencia de control ·,n·. Escribiremos los campos de datos en el mismo orden y utilizaremos el carácter de tabulación para separar los cainpos. De nuevo, como el carácter de tabulación es invisible, se representa mediante la secuencia de control"\t". Puede seleccionar cualquier delimitador lógico que resulte sencillo de distinguir. El separador o delimitador no debe ser un carácter que se utilice como entrada ya que deberla procesarse para eliminar o utilizar caracteres de escape en todas las instancias del limitador. En un capítulo posterior se analizará cómo procesar las entradas del usuario. Por el momento asumiremos que no se van a introducir tabuladores en un fon.n,ulario de pedido, Au~qµ~ result.a. difiail, no es imposible. qiie üiú|suario introduzca un tabulador o un carácter de nueva linea en un campo de entrada HTML de una sola línea. El uso de un separador es~ial de campo nos permitirá dividir los datos en variables distintas de manera más sencilla al leer los datos; como veremos en capítulos posteriores. Por el momento, trataremos cada pedido como una única cadena. Tras procesar varios pedidos, los contenidos del archivo se parecerán al ejemplo ilustrado en el listado 2.1.

Listado 2.1. Ejemplo de lo que poc:lrla contener el archivo da pedidos.

```
!5:42,   2Dtll April   e tinal oil   S aparlc plug•  $434.D0   22 Sllort St,  SUIltown
15:U,    2Dtll April   1 tires 0 oil   o •park pluga $1DD.DD   33 Main Rd,•-town
15: u,   2DtllApril    0 tirea1 oil   e apark pluga  $2&DD    127 Acacia St,
Spriagfield
```

Cerrar un archivo

Cuando se termina de utilizar un archivo, es necesario cerrarlo. Deberla hacerlo utilizando la función fcloae (1 de la siguiente forma:

```
fclose(Sfp);
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Est a e:__,ón devolv ..·· true SI e l archivo se cenó satisfactoriamente ta e:__,ón o false en caso ~on~an:·~omo la babilidad de que surja un prob ema con es ,:;: de apertura, no procederemos a probarla. es muy inferior operaad la '6 definitiva de processorder. phpse reproduce en el listado 2.2.

El código completo e ven1 n

Listado 2. • P 2 roc:eaonl•.php. Versliln definitiva de la secuencia de comandos de proceNmlento de pedldOl.

```
c?pbp
   // cree IIQllbre• de variabl~ corto•
   $tireqty • $_l'OST('tire~Y  I;
   $oilqty • $_POST('oilqty ||
   $eparkqty.  $_POST('epark~y•I;
   $addr•••.  s_POST('addr••• 1,

   $DOCUMDT_ROOT.    s_aaft11('DOCUMBIIT:..•OOT');

7.,
cbt•1.,
cbead..
   ctitle..llOb'• AUtO eart•.  order aeeultec/titte..
c/bead"
cbody..
cbi ..aob'• AUto eartec/b1..
cb2•0rder·Reeultec/b2.,
c?pbp
$date.  date(•R,i, l• r'),

ecbO '"P~~--     ~P~!•.!4*~2-- ....  -·-·  -- ..
ecbO $date,
ecbo 'c/p.,•:          ie a•  followe: c/p..•,
ecbo 'cp ..Your order

$tota~q!y:  :~ireqtY + $oilqty + $eparkqty;
!:::!ª,1t!m  ordered: , .$totalqty.•cbr  ¡..•;

if ( $totalqty •• 0)
{
   ecbo ,You did not order anytb1119 on tb• previou• pa9etcbr ¡.,•;
I
el••
(
   if | $t1reqty..o I
      ecbo $tireqty.• tireecbr ¡.,•;
   if ℓ $oilqty..0 >
      ecbo $oilqty.• bottl•• of oilcbr ¡.,;'
   if ( $epar~y..o I
      ecbO $eparkqty.• epark plugecbr ¡.,•:

$totalall0Ullt • o.oo,

def~e('TIRBPRICB',  1001;
```



```
define('OILftICS', 10);
define('SPARKftlCS', 4);

$totalamount • $tireqty • TIRBPRICS
      + $oilqty • OILPllCS
      + $eparkqty.  SPAaltftlCB,

$totalamount•llllllber_format($totala11C11111t,2,  ·•,   ·  '11

ecbo 'cp..Total of order is '.$totalamount.'c/p.,',
ecbo 'cp..llddreee to ebip to le  '.$addreee.'c/p  ..•,

$olltputetriD9 • $date.•\t•.$tireqty.•   tire•  \t•.$oilqty.•  oil\t•
      .$eparkqty.• epark pl11911\t\S".Stotala11011Dt
      .•\t•.  $addreee.•\n•;

// abra el  archivo para adjunc16n
• $fp • fopen(°$DOCIIIISIIT_ROOT/• /order•/ordere.tzt•,   'ab'I,
if  (l$fpl
{
   ecbo •cp>cetroD9.. Your order could not be proceend at tbie ti—.  '
      .'Plnee try 99ain later.c/etrong..c/p..c/bocly..c/btm:1.>'l
   exit;


fwrite($fp,  $0Utputetring,  etrlen(Soutputetringll;
fcloee($fpl;

ecbo 'cp..order vritten.c/p..•;
/..
c/body..
c/bt•l.,
```

Leer desde un archivo

Llegados a este punto, los clientes de Bob pueden remitir sus pedidos a través de la Web, pero si los empleados de Bob quieren examinar los pedidos, tendrán que abrir los archivos manualmente. Vamos a crear una interfaz Web para permitir que los empleados de Bob puedan leer los archivos de manera sencilla. El código correspondiente a esta interfaz se incluye en el listado 2.3.

Us1ado 2.3. vlewordera.php.Interfaz para leer el archNo de pedidos.

```
c?pbp
   //cree un ——n  de variable corto
   $DOCUIIBIIT_ROOT • SBTTP_snvn_vaas(.DOCCIIIS1'_11OOT'I,
?..
cbt•l..
cbead>
   ctitle..aob•• Auto Parte - Oaetomer Orderec/title..
c/bead.,
cbocly..
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
ch1>8ob'•  Auto  Part•c/111>
ch2>CU•tomer   order•c/h2>

;'":p   •  topen(•$DOCUNUT_RCIO'l'/   " /ord•r•/order•:txt•,     •r•) 1

    if  IISfpl
    |
      echo  •cp>c•tron9>•0 ordar•  pandin9.•
            .'Pl••••  try again  latar.c/•trOD9>c/p>'1
      •xit;


    whila  Ctfaof($fpll
    |
        Sordar• fg•t•l$fp,  99911
        echo $ordar.•cbr  />'1


    fcloH  (Sfpl
?>
c/body>
c/htah
```

Esta secuencia de comandos sigue el orden expuesto anteriormente: abre el archivo, lee el archivo y cierra el archivo. En la figura 2.4 se ilustra el resultado de esta ~encía de comandos utilizando él archivo de datos del listado 2.1.

FIgun 2.4. La HCUenCia de comandos YiewORlers.php muatra lodos los pedidos ac:tuÑS del ardllvo orden.txt en la ventana del navegador.

Vamos a examinar las funciones de esta secuencia de comandos en detalle.

Abrir un archivo para su lectura: fopen()

De nuevo, abrimos el archivo utilizando fopen ( ) . Eneste caso abrimos el archivo pan su lectura 'linicamente, por lo que utilizamos el modo de archivo 'r':

```
$fp.  fopanl"$DOc:UMBIIT_RCIO'l'/" /ordar•/order•,txt•,  'rb'1!
```


Saber cuéndo parar: feof()

Eneste ejemplo, utilizamos el bucle whi le para leer el archivo hasta alcanzar el final del archivo. El bucle while busca el 6nal del archivo utilizando la función feof ():

```
whil•  e, faof(C$fpll
```

La función feof () toma un puntero de archivo como su único parámetro. Devuelve true si el puntero de archivo Se encuentra al final del archivo.

En este caso (y en general al leer desde un archivo), leemos desde el archivo hasta que alcanzamos BOP.

Leer línea a línea: fgets(), fgetss() y fgetcsv()

En nuestro ejem~lo, utilizamos la función fgeu () para leer desde un archivo:

```
$0rdar•  f99t•CSfp,  99911
```

Esta función se utiliza pan leer, lfnea a línea, un archivo. En es~ caso, leerá hasta que encuentre un nuevo carácter de línea (\n), un BOP o haya leído 998 bytes del archivo. La longitud de lectura es la longitud menos un byte.

Se pueden utilizar muchas funciones diferentes pan leer uduvos. La función fgeta () resulta útil pan tratar con archivos que contengan texto plano con el que queramos trabajar en grupos.

Una variación interesante de esta función es la función fgetH (), cuya sintaxis es la sigaiente:-

```
•tring f9&t•elramource tp, int longitud, •tring (et1quatae_perm1t1d8e)I;
```

Esta función es muy parecida a la función fgeta () a,n la salvedad de que elimina todas las etiquetas PHP y HTML que encuentra. Si desea dejar alguna etiqueta en particular, puede incluirlas en la cadena ,tiqr,,ta_,.,,,.itias. Se utiliza fgetH () por seguridad al leer un archivo escrito por otra persona o que contenga entradas de usuario. La inclusión de código HTML sin restria:iones en el archivo puede estropear el formato atentamente organizado y la indusi6n de código PHP podría ofrecer rienda suelta a un usuario malintencionado sobre su servidor.

La función fgetcav () es otra variación sobre fgeta (). Su sintaxis es la siguiente:

```
array fgatcn  C ra-rc• t11,  illt 10JIQ'1tud (, etring dellldtador
        (, •triDg  cantaaedorl JI
```

Se utiliza para dividir las líneas del archivo si se ha utilizado un c:arác:tet de delimitación, como el carácter de tabulación sugerido antes o una coma como se suele hacer en las h(!jas de cálculo y otras aplicaciones. Si desea reconstruir las variables del pedido de manen individual en lugar de en fonna de lfnea de textc>,.-l"

EL_HOLISTICO@HOTMAIL.COM - 2006

fgetcsv () nos permite hacerlo con facilidad. Se invoca di la misma forma que fgets (), pero se le pasa el delimitador utilizado para separar campos. Por ejemplo:

```
Sorder • fgecc.vC$fp, 100, •\e•);
```

recupera una línea desde el archivo y la divide al encontrar un tabulador (\ t). Los resultados se devuelven en una matriz ($order en el ejemplo de código). En un capítulo posterior examinaremos las matrices.

El parámetro *longitud* deberia ser mayor que la longitud en caracteres de la línea más larga que incluya el archivo que esté intentando leer.

El parámetro *conknt dor* se utiliza para especificar por qué está encerrado cada campo en una línea. Si no se especifica, se tomarán las comillas dobles(•) como elemento predeterminado.

Leer todo el archivo: readfile(), fpassthru(), file()

En lugar de leer un archivo línea a línea, podemos leer todo el archivo de una vez. Esta operación se puede realizar de cuatro maneras.

La primera utiliza la función readfile (). Podemos sustituir la secuencia de comandos entera que escribimos anteriormente por w:ia línea:

```
readfilec•sDOCUMZIIT_ROOT/ '. /ordera/ordera.cxc•1!
```

Una llamada a la función readf i le ( ) abre el archivo, imprime el contenido de manera estándar (en el navegador) y cierra el archivo. Su sintaxis es la siguiente:

in~ ;~~dfil~(.C;i~g DOflll;~de arclliWO, -i~C [uaa~~~c:.:::l.ac:"i"iiai6a[, r880\lrce
c—cexcoJJI:

El segundo parámetro opcional especifica si PHP debería buscar el archivo en ruta inclusión y funciona de la misma forma que fopen (). El parámetro opcionar *contuto* se utiliza sólo cuando los archivos se abren de forma remota, por ejemplo con HITP, como veremos en un capítulo posterior. La función devuelve el número total de bytes leídos desde el archivo.

En segundo lugar, puede utilizar fpasstbru (). Necesitará abrir el archivo utilizando fopen () primero. A continuación puede utilizar el puntero del archivo como argumento hasta fpasstbru (), que eliminará los contenidos del archivo desde la posición del puntero hasta la salida estándar. Cierra el archivo cuando termina.

Podemos sustituir la secuencia de comandos anterior por fpasstbru () de la siguiente forma:

```
Sfp • fopenC•$0ÓCUlmllT_ROO'r/ '. /ordera/ord•r•.cxc•,  •rb'I;
fpa9ecllru.C$fpl,
```

La función fpasstbru () devuelve true si la lectura resulta satisfactoria y fal•e en caso contrario.

La tercera opción para leer el archivo entero consiste en utilizar la función file (). Esta función es idéntica a readf i le ( ) con la salvedad de que en lugar de imprimir el archivo en la aplicación estándar, lo convierte en una matriz. Examinaremos esta opción de manera más detallada en un capítulo posterior. Simplemente como referencia, se invoca utilizando el siguiente código:

```
$filaarray • fileCSDOCUMIIIT_ROOT/" /orders/orders.cxc•11
```

Esta línea de código lee el archivo entero en una matriz llamada $filearray. Cada línea del archivo se almacena en un elemento separado de una matriz. Esta función no es segura para datos binarios.

Por último, desde la versión PHP 4.3.0 puede utilizar la función file_get_contents (). Esta función es idéntica a readfile ( ). con la salvedad de que devuelve el contenido de un archivo en forma de cadena en lugar de dirigir la salida al navegador. La ventaja de esta nueva fundón es que ofrece seguridad binaria, a diferencia de la función file ().

Leer un carácter: fgetc()

Otra opción para procesar un archivo consiste en leerlo carácter a ~arácter. Para ello, se puede utilizar la función fgetc () . Esta función toma un puntero de archivo como único parámetro y devuelve el siguiente carácter de un arclBvo. Podemos sustituir el bucle wbile en nuestra secuencia de comandos ori~ por una que utilice fgetc () :

```
wbile (!feofC$fp)I
    $cbar • fgeccC$fpl;
    if !! feof ($fpll
      echo ($cbar••"\n•?  ~br />': $charl;
```

Este código lee un solo carácter del archivo a la vez utilizando fgetc () y lo almacena en la variable $cbar hasta que se alcanza el final del archivo. A continuación, sustituimos los caracteres de final de linea del texto, /n, por saltos de línea, <br [».

El único objetivo es limpiar el formato. Como los navegadores no representan una línea nueva en HTML como tal sin este código, todo el archivo se imprimirá en una línea (pruébelo). Utilizaremos el operador ternario para realizar esta operadón de manera satisfactoria.

Un pequeño efecto secundario producto del uso de fgetc () en lugar defgets ( ) es que devolvem el carácter BOF mientras que f~ets (› no lo hará. Es necesario probar feof () de nuevo tras la lectura del carácter para evitar que se imprima BOF en el navegador.

No resulta normal leer un archivo carácter a carácter a menos que queramos procesar cada uno de enos por alguna razón conaeta.

EL_HOLISTICO@HOTMAIL.COM - 2006

Leer una longitud arbitraria de bytes: fread()

Por dltimo, podemos utilizar la fundón fread (I para leer un número arbitrario de bytes de un archivo. Su sintaxis es la siguiente:

```
•tring fread(reaource t9, int lcmg1cudl;
```

Funciona de la siguiente forma: lee la longitud de bytes que se especifique o hasta llegar al final del archivo, según qui aparezca primero.

Otras funciones de archivo útiles

Existen otra serie de fundones de archivo que resultan útiles de vez en cuando.

Comprobar la existencia de un archivo: file exists()

Si desea comprobar si un archivo existe sin abrirlo, puede utilizar la función file_exists (I como se indica a continuación:

```
if (file exi9c.(o$DOC'tlMBRT ROOT/•• /orden/order..txc•11
     echo 'Tber• ar• order• waiting to be proc••••d •• ;
elae
     echo 'Tbere are c:urrently - order•. •;
```

Averiguar el tamaño de un archivo: filesize()

- La función file si.ze <I peniute comprobar él tamaño de ún ai:cliivéi Esta función devuelve el tamal\o del un archivo en bytes:

```
echo file•ize|o$DOCUIIBIIT_ROOT/ •• /order•/ord•r•.txt•I;
```

Devuelve el tamaño de un archivo en bytes y se puede utilizar en combinación con fread () para leer todo un archivo (o parte de un archivo). Podemos sustituir toda nuestra secuencia de comandos por

```
sfp • fopen(o$DOCUNIIIT_200T/ •. /order•/order•.txt•, 'rb'11
echo nl2br(fread( $fp, fileeize("$DOCUNlll'r_ROOT/ .• /order•/ordera.txt• 11;
fcloH( $fp 11
```

La función nl:Zbr convierte los caracteres \n del resultado en saltos de linea HTML (cbr />).

Eliminar un archivo: unlink()

Si desea eliminar un archivo tras procesar los pedidos, puede hacerlo con ayuda de la funádn unl ink (). (No existe una función llamada delete.) Por ejemplo:

```
unlinlc(o$DOCUMDT_ROOT/ " /ordera/ordar•.txt•I:
```


Esta función devuelve false si el archivo no se puede eliminar, lo que suele ocurrir si los permisos asociados al archivo no resultan suficientes o si no existe el archivo.

Desplazarse dentro de un archivo: rewind(), fseek() y ftell()

Puede manipular y descubrir la ubicación del puntero del archivo dentro de un archivo utilizando rewind (), fseek () y ftell ().

La función rewind (I restablece el puntero del archivo al comienzo del archivo. La función ftell (I indica la distancia a la que se encuentra el puntero dentro del archivo en bytes. Por ejemplo, podemos agregar las siguientes líneas a la parte finaJ de nuestra secuencia de comandos (antes del comando fe lose ()).

```
echo 'Final po•ition of the file pointer i• •.(ftell(Sfpll;
ecbo 'cbr />';
rewind($fpl;
echo 'After rewind, tbe po•ition 1• •.(ftell($fpll,
echo 'cbr />';
```

El resultado del navegador será similar al ilustrado en la figura 2.5.

Figura 2.5. Tras leer los pedidos, el puntero apunta al final del archivo, una distancia de 234 bytes. La llamada de retroceso lo establece a la poslci6n 0, es decir al principio del archivo.

La función fseek () se puede utilizar para establecer el puntero del archivo en otro punto dentro del archivo. Su sintaxis es la siguiente:

```
int faeek < reao.arc• tp, int deapluamuco 1, 1nt JNDCO de ,,;rt1ül I
```

Una llamada a f~eek () establece el puntero fp en un punto desde los que desplaza los bytes indicados. El parámetro opcional *punto* dt p,rlill• adopta como

EL_HOLISTICO@HOTMAIL.COM - 2006

valor predeterminado SEEK_SET que es el principio del archivo. Los otros valores posibles son SEEK_CUR (la ubicación actual del puntero del atthivo) y SEU:_END (el final del archivo). La función rewind equivale a llamar a la función faeek con un desplazamiento 0. Por ejemplo, podemos utilizar f seek ( ) para buscar el registro intermedio de un archivo o para realizar una búsqueda binaria. A menudo, si alcanza el nivel de complejidad de un archivo de datos en el que necesite realizar este tipo de tareas, le resultará mucho más sencillo utilizar una base de datos.

Bloquear archivos

Imagine una situación en la que dos dientes intenten realizar un pedido de un producto a la vez. (Algo que no resulta extraño, en especial si el volumen de tráfico del sitio Web comienza a tener cierta importancia.) Piense en qué ocurriría si un cliente llama a la función fopen ( ) y comienza a escribir, y entonces otro cliente llama a la misma función y también escribe. ¿Qué contendrá al final el archivo? ¿Será uno de los pedidos o el otro? O algo menos útil, como dos pedidos entremezclados. La respuesta depende del sistema operativo, pero a menudo resulta imposible de saber. Para evitar problemas como éste, se puede utilizar el bloqueo de archivos. Para implementar este recurso en PHP se utiliza la función flock (>.Esta función debería invocarse tras abrir un archivo, pero antes de que se lea o escriba ningún dato. Su s~taxis es la siguiente:

```
bool Uock(rHoarce tp, inc opencl da  |, inc 6bloquNr•J  I
```

Es necesario pasarle un puntero al archivo abierto y un número que represente el tipo de bloqueódeseado. Devuelve true si el bloqueo se logra satisfactoriamente y false si no es así. El tercer parámetro opcional contendrá el valor *tnlt* si el bloque se puede aplicar al proceso actual (es decir, hacer que éste espere).

En la tabla 2.2 se recoge un resumen con los valores posibles del parámetro *operación*. Estos valores cambiaron en la venión 4.0.1 de PHP. En la tabla se recopilan ambos conjuntos de valores.

Tabla 2.2. Valores de operación de la función llock().

Valor de operación	Significado
LOCK_SH (antes 1)	Bloqueo de lectura. El archivo se puede compartir con oCros lectores.
LOCK_EX (antN 2)	Bloqueo de escritura. Es exclusivo. El archivo no se puede compartir.
LOCK_UN (antes 3)	Uberar bloqueo ulstente.
LOCK_NB (antes 4)	SI se agrega el número 4 a la operacl6n, se Impide el bloqueo al intentar obtener uno.

Si va a utilizar f lock ( ), tendrá que agregarlo a todas las secuencias de comandos que utilicen el archivo ya que de lo contrario carecerá de valor.

flock (> no funciona con el sistema de archivos NFS ni con otros sistemas de archivos de red. Tampoco se puede utilizar con sistemas de atthivos antiguos que no admitan bloqueos como FAT. En algunos sistemas operativos se implementa en el nivel de proceso y no funcionará correctamente si utiliza un API de servidor multiproceso.

Para utilizarlo en el ejemplo, es necesario modificar proceHorder. php de la siguiente forma:

```
$fp • fopen("$DOCONSIIT_II00T/·. /order•/order•.cxc•,  •ab'I;
flockl$fp,  LOCJt_BXI; // bloquee el archivo para •u e•cricure
fvricel$fp,  $oucpuc•tringl;
flockl$fp, LOCX_llllll    // de•accive el bloqueo de e•critura
fcloH($fpl;
```

Tambim debería agregar bloqueos a viewordera. php:

```
$fp • fopenl"$DOCUNIIIT_ROOT / •• /orcler•/ordMire.cxc•,  'r'I;
flock($fp,  LOCX_SRI;   // bloquee el archivo para lec:cura
// lH  de.Se el archivo
flock($fp,  LOCX_llllll;   // cleeaccive el bloqueo de leccura
fclOH ($fpl
```

Nuestro código resulta ahora más sólido pero no es perfecto. ¿Qué ocurrirá si dos secuencias de comandos intentan obtener un bloqueo a la vez? Que se creará una carrera en la que los procesos competirán por conseguir bloqueos sin saber quién lo logrará primero, lo que puede ser fuente de más probl~ Lo mejor es un utilizar un DBMS.

La opción más acertada: los sistemas de administración de base de datos

Hasta el momento todos los ejemplos vistos utilizan archivos planos. En la siguiente sección examinaremos cómo utilizar MySQL,un sistema de administración de bases de datos relacional. Es posible que se esté preguntando por qué molestarse en recurrir a este sistema.

Problemas con el uso de archivos planos

El uso de archivos planos plantea una serie de problemas:

- Cuando un archivo supera un determinado tamaño, el trabajo se ralentiza.
- La búsqueda de un determinado registro o de un grupo de ellos resulta difícil dentro de un atthivo plano. Si los registros están ordenados, puede utilizar

EL_HOLISTICO@HOTMAIL.COM - 2006

algón tipo de búsqueda binaria en combinadón con un registro de ancho fijo para buscar sobre un campo da,-e. Si desea buscar patrones de informadón (por ejemplo, para extraer todos los clientes que viven en una determinada dudad), tendrá que leer cada registro y comprobarlo de manera individual.

- Los accesos simultúteos pueden resultar difíciles de soludonar. Ya hemos visto cómo bloquear archivos, pero este mecanismo puede dar lugar a una carrera entre procesos. Tambim pueden dar lugar a cuellos de botella. En determinados niveles de tráfico, puede ocurrir que un grupo grande de usuarios tenga que esperar a que se desbloquee el archivo para poder realizar un pedido. Si la espera es muy larga, se irán a comprar a otra parte.
- El procesamiento de archivos que hemos visto hasta el momento es de tipo secuencial, es decir, partimos del principio del archivo y lo leemos progresivamente hasta llegar al final. Si queremos inserta!' o eliminar registros de la parte central del archivo (acceso aleatorio), la opención resultará complicada ya que será necesario leer todo el archivo en memoria, realizar los cambios y volver a escribir el archivo completo. Si el archivo de datos es grande, la sobrecarga será significativa.
- Aparte de las funciones que ofncen de los permisos de archivos, no existe una forma sencilla de aplicar diferentes niveles de acceso a datos.

Resolver estos problemas con RDBMS

Los sistemas de administradón de bases de datos dan una respuesta a todos estos problemas:

- Los RDBMS proporcionan acceso más rápido a los datos que los archivos planos. MySQL, el sistema de base de datos que utilizaremos en este libro, es uno de los más rápidos.
- Resulta sencillo consultar los RDBMS para extraer conjuntos de datos que se correspondan con determinadas criterios.
- Los RDBMS incorporan mecanismos integrados para resolver el problema de los accesos simul~eos para que los programadores no tengan que preocuparse.
- Los RDBMS propordonan acceso aleatorio a los datos.
- Los RDBMS incorporan sistemas de privilegios. MySQL dispone de funciones especiales en este sentido.

Probablemente, la razón principal para utilizar un RDBMS es que implementan todas las funciones (o al menos la mayor parte de ellas) que se desean para un sistema dé almacenamiento. Puede escribir su propia biblioteca de fundones de PHP, pero ¿por qué tomarse esa molestia?

En la segunda pmte de este libro, se analizará el funcionamiento general de las bases de datos y cómo configurar MySQL para crear sitios Web dotados de bues de datos.

Si tiene pensado crear un sistema sencillo y cree que no necesitari una bae d datos completa para desea evitar el bloqueo y otros problemas asodados a .0. archivos planos, puede qu~ le interese la nueva extensión SQLite de PHP. Msicamente lo que ofrece es una mterfaz SQL a un archivo plano. En el h'bronos centraremos l;l' el uso de MySQL pero si necesita más información sobre SQLite no dude en acudir a http://aqlite.org y http://www.php.net/sqlite.

Lecturas adicionales

En un capítulo poelerior veremos cómo intenctuar con los sistemas de archivos. En concreto, se indicará cómo cambiar permisos, derechos y nombres de archivos có~ trab~jar con directorios y cómo intenctuar con el entorno del sistema d; archivos.

También puede consultar la sección dedicada al sistema de archivos del ~ual en línea de PHP ~n bttp: / /www.pbp.net/ filesystem.

A continuación

En el siguiente capítulo, analizaremos las matrices y cómo utilizarlas para procesar datos en secuencias de comandos PHP.

EL_HOLISTICO@HOTMAIL.COM - 2006

3

Utilizar matrices

En este capitulo se explica cómo utilizar una importante estructura de programación: las matrices. Las variables que hemos visto en los capítulos anteriores sólo almacenan un valor. Una matriz es una variable capaz de almacenar un conjunto o secuencia de valores. Una matriz puede constar de una gran cantidad de elementos. Cada elemento puede albergar un único valor, como texto onúméros.-u otra matriz. Las matrices que contienen otras matrices se conocen como matrices multidimensionales. PHP admite matrices indexadas numéricamente y matrices asociativas. Es probable que esté familiarizado con las matrices indexadas numéricamente si ha utilizado algún lenguaje de programación, pero si no ha programado nunca en PHP o en Peri es muy probable que no conozca las matrices asociativas. Estas matrices permiten utilizar valores m4s útiles como indice. Las matrices asociativas permiten asociar a cada elemento palabras u otro tipo de información con significado en lugar de indices numéricos. En este capítulo retomaremos el ejemplo de la aplicación Bob's Auto Parts y utilizaremos matrices para facilitar el trabajo con información repetitiva como los pedidos de los clientes. Así mismo, escribiremos código más breve y claro para realizar parte de las tareas desarrolladas en el capítulo anterior. Nos centraremos en los siguientes aspectos:

- Matrices indexadas numéricamente
- Matrices indexadas no numéricamente
- Operadores de matriz
- Matrices multidimensionales

EL_HOLISTICO@HOTMAIL.COM - 2006

- Ordenar matrices
- Funciones de matriz

¡Qué es una matriz!

En un capitulo anterior vill\OS las variables escalares· Una variable escalar es una ubicación con nombre en la que se almacena una variable; de manera similar, un matriz es una ubicación con nombre para atmacenar un conjunto de valores, que, por tanto, permite agrupar variables escalares.

Utilizaremos la lista de productos de Bob para crear la matriz de nuestro ejemplo. En la figura 3.1 se puede ver una lista de tres productos almacenados en formato de matriz en una variable, denominada $products, que contiene tres valores. (En un in&tanteexplicaremos cómo crear un variable como ésta.)

Figura 3.1. Los producto• de Bob se pueden almacenar en una matriz.

Tras colocar la información en una matriz, podemos trabajar con ella de varias formas útiles- Si utilizamos la estructura de bucle del capítulo 1, podemos ahorrar trabajo aplicando las mismas acciones a cada valor de la matriz. Podemos trasladar todo el conjunto de información de un sitio a otrOcomo si se tratara de una unidad. De esta forma, bastarA con una- sola-llnea-de.có.di8º ~!ª pasar todos los valores a una función. Por ejemplo, podriamOS ordenar todos los productos alfabéticamente pasAndoselos a la función sort C)· Los valores almacenados en una matriz se denominan elementos de matriz- Cada elemento lleva asociado un índice (también denominado clave) que se utiliza para acceder al elemento. Las matrices de la mayor parte de los lenguajes de programación constan de índices numéricos que suelen comenzar en cero o en uno. PHP le permite utilizar números o cadenas como índices. Puede utilizar matrices indexadas de la forma numérica tradicional o establecer las claves en lo que desee para que los indices tengan mú sentido y le resultan mú útiles- El enfoque de programación puede variar en función de si utiliza matrices estú\dar indexadas numéricamente u otros valores de indice mú interesantes. comenzaremos por e,caminar las matrices indexadas numéricamente y, tras ello, pasaremos a las claves definidas por el usuario.

Matrices indexadas numéricamente

La mayor parte de los lenguajes de programación permite el uso de las matrices indexadas numéricamente. En PHP, los índices comienzan en cero de manera predeterminada pero se puede variar este parámetro·

Inicializar matrices indexadas numéracamente

Para crear la matriz ilustrada en la figura 3.1, utibce la siguiente línea de código de PHP:

```
$procluct• • array( 'Ti re•·· ·011•, •spsrk Plug•• I·
```

Esta linea de código crea una matriz llamada 'a producta que contiene los valores dados 'Tires', 'Oil', 'Pluga' · Como en el caso de la función echo, arra y () es más una~ ... ms: ... olgulen.. ,j,mplo. Todo~ loo ..... de... ~-...-. Puede que no necesite inidaliz tructural del lenguaje que una fun... a os que necesita en otra tri conterudos de la matriz. S: Si quiere abna ma z, puede copiarlos utilizand 1 i tiene utilizar la fundócenar una secuencia ascendente de núme o e operador '". línea de código n ra,nge(1 para crear la matriz automá en una matriz puede aear una matriz den mente. La siguiente 10 pnmeros números. ominada numbera cuyos elementos son los

```
Snllllber• • ran,ge(l,101;
```

La función range ( ) tiene un terc á ~e.r el tamaño intermedio entre valo':s:8:om:;ro~pci~l que le permite establewneros pares entre 1 y 10, podría crearla de ~.t.Z.. desea una matriz con los

```
Sodd• • ran,ge(l. 10, 21;
```

La fundónrange() tambi"én se puede utilizar con caracteres , como se indica a continuación.

```
$letter• • range , •, , , z,
```

Si tiene la info 6 los contenidos en un archivo del disco duro, pued e cargar $: l d atos destinados a la matriz vo como v eremos en un sección posterior ~ ...~ -:,, :;~usposc~~dos directamente d.:: la ==:ee:.':'8 base de datos, pu~e enor. os, como veremos en un

También puede utilizar varias f unación para extraer parte de una matriz o para volver a ordenarla. En una secc1 n posterior veremos parte de estas fundones.

Acceder a los contenidos de matrices

Para acceder a los contenid os de una matriZ utilice La la iz, u ce el nombre de variable o el ~d su nombre. Si la variable es I c ve o el índice indican a qué valores alma1ce~ acceder a los contenidos. co oca 4:l'lre corchetes tras el nombre. cena os accedemos. El fndice se E9CDba$producta [O] $ rod tenidos de la matriz de prc:d!:tos.ucts [11 y$producta [21 para utilizarlos con-,

EL_HOLISTICO@HOTMAIL.COM - 2006

El primer elemento de la matriz es el elemento cero. Se trata del mismo sistema de numeración utilizado en C, C++, Java y otros lenguajes, pero puede que al principio le cueste acostumbrarse a 11.

Como ocurre con otras variables, para cambiar los contenidos de los elementos de matriz se utiliza el operador •· La siguiente lfnea sustituirA el primer elemento de la matriz, •Tires•, por •Puses•.

```
Sproduct•(0J  • 'Fu•••';
```

La linea que se incluye a continuación podría utilizarse para agregar un nuevo elemento (• Puses ') al final de la matriz, lo que nos da un total de cuatro elementos:

```
Sproduct• l3 J  • 'ru.e• ';
```

Para mostrar los contenidos, podríamos escribir:

```
echo  •sproduct•I0J  Sproduct•Cll  Sproduct•(ZJ  Sproduct•[J)•;
```

Tenga en cuenta que aunque el anAlisis de cadenas que realiza PHP es bastante inteligente, puede que resulte un poco confuso. Si tiene problemas porque las matrices u otras variables no se interpretan correctamente al encerrarlas entre comillas dobles, puede colocarlas fuera de las comillas. La instrucdónecho anterior fundonarA correctamente, pero en gran parte de los ejemplos posteriores, de mayor complejidad, no se utilizan cadenas con comillas.

Como ocurre con otras variables de PHP, no es necesario inicializar las matrices o crearlas por adelantado ya que se crean automAticamente la primera vez que se utilizan.

El-sigÜiC!ñte c:&11gc,-¿re¡r,1. misma matnz-sproaücts:

```
Sproduct•l0I   •  'Tire••;
Sproduct•Cll  •  '0il';
Sproduct•lZI  •  'spark  Pluga:
```

Si la matriz $products no existiera todavía, la primera linea crearla una nueva matriz con un único elemento y las siguientes líneas agregarian valores a la matriz. El tamaii.o de la matriz cambia automAticamente al 81\adir elementos a la misma, posibilidad que no se incluye en otros lenguajes de programación.

Utilizar bucles para acceder a la matriz

Como la matriz se indexa a partir de una secuencia de números, podemos utilizar un bucle for para mostrar los contenidos de manera mú sencilla.

```
for ~,,U • 0;  $~c3;  $1++ I

   ecllo °$product•($i)   •;
```

Este bucle produce resultados similares al código anterior, pero exige menos trabajo ya que no es necesario esaibir código para trabajar con cada elemento de

una matriz. Una de las características destacadas de las matrices indexadas • a: posibilidad de utilizar un sencillo bucle para acceder a cada uno de sus eJemmtm. En el caso de matrices asociativas no resulta tan sencillo utilizar bucles para recorrerlas, pero permiten usar índices con valores significativos. Tambiffl pode,,, mos recurrir el bucle foreach que estA espec:ialmente diseftado para su uso can matrices. En este ejemplo podemos utilizarlo de la siguiente forma:

```
foreach jSproduct•  ••  Scurreatl
   echo  Scurreac.'   ';
```

Este código almacena elemento a elemento en la variable $current y los imprime.

Matrices con diferentes índices

En la matriz de productos, permitimos que PHP asigne a cada elemento el índice predeterminado. Esto significa que el primer elemento que agregamos se convierte en 0, el segundo en 1 y así sucesivamente. PHP admite el uso de matrices en lu que a cada valor podemos asociar cualquier clave o índice.

Inicializar una matriz

El siguiente código crea una matriz utilizando los nombres de productos y los precios como valores.

```
Spric••  •  array(  'Tir••'•>100,   '0il'•>10,   •spark Plug•'•>4   li'
```

El símbolo entre claves y valores es un sendllo signo igual seguido de un símbolo mayor que.

Acceder a elementos de matriz

De nuevo, utilizamos el nombre de la variable y una clave para acceder a los contenidos. De esta forma podemos acceder a la información almacenada en la matriz de precios como $prices (•Tires• l, $prices (•Oil • J y $prices ['Spark Plugs ' l . El siguiente fragmento de código crea la misma matriz $prices. En lugar de crear una matriz con tres elementos, esta versión crea una matriz con un único elemento y, a continuación, crea dos mú.

```
Spric••  •  array(  'Tir••'•>100   J;
Sprice•('0il'J     •  10;
Spric••l'Sperk   Plug••J  •  4;
```

Aquí tambim existe una pequei\a diferencia, aunque el código sea equivalente. En esta versión, no creamos una matriz de manera explicita sino que se crea automAticamente al agregarle el primer elemento.

EL_HOLISTICO@HOTMAIL.COM - 2006

```
$pric••l'T1r••'I  • 100;
Spric••l'Oil'I   • 10!
$prlce•l'Spark  Plug•'I  • 4;
```

Utilizar bucles

Como los índices de esta matriz no estút formados por números, no podemos incluir un simple contador dentro del bucle for para que funcione con la matriz. Podemos utilizar el bucle foreach o las instrucdones list (I y each ().

El bucle foreach presenta una estructura ligeramente diferente cuando se utiliza con matrices asociativas. Podemos usarlo tal y como hicimos en el ejemplo anterior o incorporar las claves:

```
foreach CSpric•• •• Skey •> svaluel
   echo Skey. '•>' .$Value. •cbr />' 1
```

El siguiente código devuelve los contenidos de la matriz $prices utilizando la instrucción each (I :

```
while( sei—t • each( Spric•• | >
(
   echo Sele11e11t( •key' I;
   echo • - •/
   edlo Sele11e11tl •value• II
   echÔ ·cbr />';
```

El resultado de este fragmento de código se muestra en la figura 3.2.

Figura 3.2. Se puede ullllzar una in1trucci6n uch() par11 recorrer matricu:

En un capitulo anterior, vimos los bucles while y la instrucción echo. El c6digo anterior utiliza la función eacb (I, que no se ha visto todavía. Esta función d«!'"Uelve el elemento actual de la matriz y convierte al siguiente elemento en el actual. Como esÍamos lla.maJ\doa eacb (1 dentro de un bucle wbile, los elementos de la matriz se devolverút uno a uno y se detendrá al alcanzar el final de la matriz.

En este c6digo, la variable $element es una matriz. Al llamar a eacb (). nos devuelve una matriz con cuatro valores y los cuatro índices hasta las ubicaciones

de matriz. Lu ubicaciones key y o contienen la clave del elemento actual y las ubicaciones value y 1 contienen el valor del elemento actual. Aunque da igual cuál se escoja, en nuestro ejemplo hemos optado por utilizar las ubicaciones con nombre en lugar de las ubicaciones con número.

Existe una forma m6s habitual y elegante de realizar esta tarea. Podemos utilizar la función list () para dividir una matriz en un número de valores y separar I~ dos valores que nos proporciona la función eacb ( ) de la siguiente forma:

```
Sl1•tC SpEOduct, $price 1 • each( $pr1ce• I;
```

Esta línea utiliza eacb ( ) para tomar el elemento actual desde $prices, lo devuelve en forma de una matriz y convierte al siguiente elemento en el actwll. También utiliza la función 1 ist () para convertir los elementos 0 y 1 de la matriz devueltos por each () en dos nuevas variables llamadas $product y $price.

Podemos utilizar un bucle para recorrer la matriz $prices entera y"mostrar los resultados utilizando esta breve secuencia de comandos.

```
wbil• ( li•tC Sproduct. $price | • each( Sprice• | |
   echo •Sproduct - Spricecbr />";
```

El resultado de este código es el mismo que el generado por la secuencia de comandos anterior, pero resulta m6s sencillo de leer porque la instrua:ión 1 ist () permite asignar nombres a las variables. Una cosa en la que fijarse al utilizar each () es que la matriz realiza el seguimiento del elemento actual. Si qu——os utilizar la matriz dos veces en la misma secuencia de comandos, debemos restablecer el elemento actual al principio de la matriz utilizando la función reset <) , Para recorrer la matriz de precios de nuevo con ayuda de un bucle, utilice el siguiente código:

```
AHt   ($priCH)   ;
while ( li•tl $produce, $price | • eacbC $price• | |
   echo "SpEOduct • Spricecbr />";
```

Este código restablece el elemento actual al principio de la matriz y nOS permite recorrerla de nuevo.

Operadores de matriz

Existe un conjunto especial de operadores que se aplican a las matrices. La mayor parte de ellos cuentan con un operador escalar hom6logo, como se puede comprobar en la tabla 3.1. Estos operadores son bastante evidentes pero el proceso de unión requiere una derta explicación. El operador de unión intenta aiiadir los elementos de $b al final de $a. Si los elementos de $b tienen las misinas clavesque algunos elementos ya presentes en $a, no se alladirán. Es decir, no se sobl'esc:ril,irút elementos de $a.

Comprobará que todos los operadores de matriz de la tabla anterior cuentan con operadores equivalentes que funcionan en variables escalares. Siempre que reé:uer·

EL_HOLISTICO@HOTMAIL.COM - 2006

de que el operador+ realiza sumas en tipos escalares y uniones en matrices, aunque no le interese la aritmética de la operación, los comportamientos tendrán sentido. No se pueden comparar las matrices a los tipos escalares.

rabia 3.1. Opermdorea da matriz de PHP.

Operador	Nombre	Ejemplo	Resultado
+	Unión	$a+ $b	Unión da $• y $b. La matriz $b se adjunta a $a, paro no se aftaden In collslonas da cla1a1.
	Igualdad	$a •• $b	True si $a y $b contienen lol mllmos alamantos.
	Identidad	$a ••• $b	True li $a y $b contienen lol mismos alemantos en el mismo orden.
!•	Dasigualdad	$a !• $b	True li $a y $b no contienen los mismos elementos.
<>	Dasigualdad	$a <> $b	Igual que !••
!••	No idantic:lad	$a !•• $b	True si $a y $b no contienen los mismos elementos en et mismo orden.

Matrices multidimensionales

Las matrices no tienen por quf estar formadas por una simple lista de claves y valores ya que cada ubicación de la matriz puede contener otra matriz. De esta forma podemos crear una matriz de dos dimensiones. Para hacene una idea del concepto de las matrices bidimensionales, piense en ellas como una tabla, o cuadrícula, con altura y anchura o con filas y columnas.

Si queremos almacenar más de un dato de cada producto de Bob, podemos utilizar una matriz bidimensional. La figura 3.3 muestra los productos de Bob en una matriz bidimensional en la que cada fila representa un producto y cada columna un atributo de producto. Utilizando PHP podrfamos escribir el siguiente código para configurar los datos de la matriz que se muestra en la figura 3.3.

```
$producta • arrayC arrayC 'TIR', 'Tirea•, 100 l,
        arrayC •oIL', 'Oil •, 10 I.
        arrayl 'SPIC', •spark Pluga•, 4 1 );
```

En~ta definid~n puede ver que la matriz de productos contiene aban tres matrices.

Para acceder a los datos de una matriz unidimensional, recuerde que necesitamoa el nombre de la matriz y el índice del elemento. En una matriz bidimensional la

operación es similar con la salvedad de que cada elemento consta de dos índices, una fila y una columna. (La fila superior es la fila 0 y la columna situada mú a la izquierda es la columna O.)

:~~i,;;	~	;f,~\ ..
nR	1'w	!00
OIL	OI	10
SPI(	Spmkl'!!III	4

tl

Figura 3.3. Podamos almacenar m61 Información sobre los productos de Bob en una matriz bidimensional•

Para visualizar los contenidos de esta matriz, podríamos acceder manualmente a cada elemento de la siguiente forma:

```
echo '!' .$produc:ɯ[0) [DI.' !' .$produc:ta[OJ (1). • ! • .$produc:ta(0J (21. • !cbr />' I
ec:ho '[' .$produc:ta(1) [0) .'1'.$produc:ta(1) (1).'1'.$produc:ta(1) (2).'lcbr />';
ec:ho '!' .$produc:ɯ[2) [DI.' ! '.$produc:ta(2J (1). '.|' .$produc:ta(2] [t]. 'lcbr />';
```

Tambim podemos colocar un bucle for dentro de otro bucle fo,: para obtener el mismo resultado.

```
for | $row • 0; $row e 3; $row++)
( -'-  -  ---'-'-'-—-—-—-'-'-—-  -  -
   for ( $C:Ol111111 • 0; $c:olumm e 3; $c:ol111111++
   (
      ec:ho '!'.$praducta($~J [$col11111DI;

   echo 'lcbr />' I
```

Ambas versiones generan el mismo resultado en el navegador.

```
ITIRITirea110DI
!OIL!Oil;10!
ISPXl&park Pluga!4!
```

La única diferencia entre los dos ejemplos es que el código ser6 mú breve si utiliza la segunda versión con una matriz de gran tamaJlo.

Si lo prefiere puede crear nombres de columna en lagar de números, como se ilustra en la figura 3.3. Para ello, puede utilizar matrices asociativas. Para almacenar el mismo conjunto de productos, con columnas con nombre como las de la figura 3.3, se utiliza el siguiente código:

```
Sproclucta • arrayC .array! 'Coda' •> 'Tl11',
                            'De•cription' •> 'Tirea•,
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
        •trice•  •> 100
    ,'
    array(  'Coda'  •> 'OIL',
            'DaacriptiOD'  •> '011•,
            '•rica•  ... 10
    ,'
    array(  'Coda'  •> 'SPIC',
            'Daacription•  .,.  ·s~k   Plw¡a•,
            •11rïce•  ••4
    |
!;
```

Resulta mú sencillo trabajar con esta matriz si sólo se desea recuperar un valor y es más sencillo recordar que la descripción se almacena en la columna De•cription que recordar que estj almacenada en la columna 1. El uso de indices descriptivos evita tener que memorizar que se &a almacenado un artfculo en [x)(yJ. Es fácil encOl\trar los datos utilizando la referencia a una ubicación por medio de nombres de fila y columna significativos.

Sin embargo, perderemos la posibilidad de utilizar un sencillo bucle for para recoirer cada columna. Una forma de escribir código pua mostru esta matriz es la siguiente:

```
for t Srow • o,  Srow e 3;  Srow++ >
{
   ac:J1o  •1•.sproducta(Srowl   ('Coda').'1',$producta(Srow]('Da•criptioa•J.
        •1•.$proctucta(Srowl  ('Prica•).'lcbr  />'1
```

EJ'uso de un bucle for permite recorrer la matriz externa $produce., nummcamente-indexada. Cada fila deJa.matriz .$prod~cta es un Jllatriz asociativa. Podemos utilizar las funciones each () y liat () en un bucle w!liíe para recorrer matrices asociativas. Por lo tanto, necesitamos utilizar un bucle while dentro de un bucle for:

```
for C  $row • O;  $row e 31  Srow++ 1
{
   wllila I liatl  Skey,  Svalua l • eac:hl $producta( $row J l l
   {
      acbo "l$Yalua•1
   )
   ecbo  'lcbr  />' ,
```

No tenemos por que! limitamos a dos dimensiones. De la misma forma que los elementos de las matrices pueden contener otras matrices, esas nuevas matrices pueden contener a su vez otras. Una matriz tridimensional tiene altura, anchura y fondo. Si el súnil ~e las matrices bidimensionales como tab]as con filas y columnas le •f11'!6 a imaginárselas, piense en las matrices tridimensionales como un c:anjunto •Jñiro de tablas. Para hacer referencia a cada elemento, se utilizará la capa, la fila y la columna. Si Bob dividiera sus productos en categorias, podrfamos utilizar una matriz tridimensional para almacenarlos. La figura 3.4 muestra los productos de Bob en una matriz tridimensional.

FIgun 3.4. Esta matriz tridimenslonal nos pemite dividir toe productos en c:at8gorl~.

Si examina el código que define esta matriz, observará que una matriz tridimensional es uña matriz que contiene matrices de matrices.

```
$cat990riea • arrayl erray I arrayl  'CAII_TIR',  'Tiraa•,  100 ),
                             array(  'CAII_OIL',  'Oil',  10 ),
                             arrayl  'CAR_SPlt', 'spark  Plaga',  4
      ),
   array I arrayl  'VAII_TIR',  'Tiraa•,  120 l,
           arrayl  'JfAII_OIL',  'Q11•,  12 ,.
           array(  'VAII_SPIC',  '&park  jiuga;,--5  i" --
      ,'
   array I arrayl  •TRJt_TIR',  'Tiraa•,  150 l,
           arrayl  'TRlt_OlL',  'Oil',  15 ),
           array(  'TRlt_SPIC', 'Spark  Plaga•,  ,
      )
   );
```

Como esta matriz sólo consta de fndices nummcos, podemos utilizar bucles for anidados pmra mostru sus contenidos.

```
for I Slay.r • O;  $layez < 3;  $layar++ l
(
   ecbo •Layar  $layercbr />";
   for ℓ $row • 0,  $row e 31  $row+• 1
   (
      for  ( $Colimm • O;  $collmll e 31  $C01WIII++
      (
         ecbo  •1•.scategoriaa($layer]($rowl($columal;
      }
      echo  'lc:br  />';
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Debido a la forma en la que se crean las matrices multidimensionales, podríamos crear matrices de cuatro, cinco o seis dimensiones. El lenguaje no tiene límites a este respecto, pero resulta dificil visualizar estructuras con más de tres dimensiones. La mayor parte de los problemas del mundo real se corresponden lógicamente con estructuras de tres o menos dimensiones.

Ordenar matrices

A menudo resulta útil ordenar datos relacionados que estm almacenados en una matriz. Es sencillo tomar un matriz unidimensional y ordenarla.

Utilizar sort()

El siguiente código da como resultado una matriz que se ordena en orden alfab4tico ascendente:

```
Sproduct• • array! 'Tire•', 'Oil', 'Spark Plug•' I;
•ort(Sproduct•I;
```

Los elementos de nuestra matriz quedarán ordenados de la siguiente forma: 011, Spark Pluga, Tires.

Tambim podemos ordenar los valores numéricamente. Si tenemos una matriz con los precios de los productos de Bob, podemos ordenarla en sentido numérico ascendente como se muestra a continuación:

```
$pric:;.-;; arrav<"" iiiéi;"" 10, 4 1.;
•ort(Sprice•I;
```

Los precios se ordenarán ahora de la siguiente forma: 4, 10, 100.

Tenga en cuenta que la función sort (l discrimina entre mayúsculas y minúsculas, y que las letras mayúsculas van delante de las minúsculas. Por lo tanto, la "A" va antes que "Z', pero la "Z' va antes que la "a".

La funáón tambim cuenta con un segundo parámetro opcional. Puede pasar una de las siguientes constantes: SORT_RBGULAR (la predeterminada), SORT_NUMERIC o SORT_STRING. La posibilidad de especificar el tipo de ordenación resulta muy útil para comparar cadenas que puedan incluir números, por ejemplo, 2 y 12. Numéricamente 2 es menor que 12 pero si se trata de cadenas '12' es menor que '2'.

Utilizar asort() y ksort() para ordenar matrices

Si quisiéramos utilizar una matriz asociativa pua almacenar artículos y sus precios, necesitaríamos utilizar diferentes tipos de funciones de ordenadón para mantener agrupadas las claves y su valores al ordenarlos.

El siguiente código crea una matriz asociativa en la que se incluyen tres prodw» tos y sus precios asociados, y ordena la matriz en orden ascendente por los precios.

```
$prics•. arrayC 'Tire•'•>100, '0il'•>10, 'spark PlUCJ•'•>4 );
••ort($price•I;
```

La función asort () ordena la matriz en función del valor de cada elemento. En la matriz, los valores son los precios y las claves son las descripciones textuales. Si en Jugar de ordenar la matriz por el precio, queremos hacerlo por la descripción, se utilizará la función ksort. (l, que ordena la matriz por la clave y no por su valor. Como resultado se ordenarán las claves de la matriz allabéticamente: Oil, Spark Plugs, Tires.

```
$pric••. array( 'Tire•'•>100, '011'•>10, 'spark Plug•'•>4 );
ıortl$pricH!l
```

Invertir el orden

Hemos visto las funciones sort (l, asort (l y ksort (l. Estas tres funci~es ordenan Ju matrices en orden ascendente. Pero cada una de ellas tieQJ una función correspondiente para ordenar las matrices en orden descendente. Estas funciones inversas son rsort (), arsort () y krsort (). Las funciones de ordenación inversa se utilizan de la misma forma que las funciones de ordenación normales. La función rsort () ordena un matriz unidimensional indexada numé.ricamente en orden descendente. La función arsort () ordena una mat~ asociativa unidimensional en sentido descendente utilizando el valor de cada elemento. La función krsort (l ordena una matriz asociativa unidimensional en tlrden deseendente utiizañdó la_clave de cáda éleirierito.

Ordenar matrices multidimensionales

La o~enación de matrices con más de una dimensión o aplicando un orden distinto al alfab4tico o numérico resulta más complicado. PHP sabe cómo comparar dos números o dos cadenas de texto, pero en una matriz multidimensional cada elemento de una matriz es otra matriz. PHP no sabe cómo comparar dos matrices, por lo que es necesario aear un método para compararlas. En la mayor pute de las ocasiones, el orden de las palabras o números resulta bastanje obvio pero en el caso de objetos complicados, se convierte en una operación más problemática.

Ordenaciones definidas por el usuario

A continuación se repite la definición de una matriz bidimensional utilizada anteriormente. Esta .matriz almacena los tres productos que comercializa Bob con un código, una descripción y un precio.

EL_HOLISTICO@HOTMAIL.COM - 2006

```
Sproducc•.  arrayC array( 'TIR', 'Tire•'·  100 1,
                  arrayC 'OIL', 'O11',   10 ),
                  arrayC 'SPlt'.   •spartc  Plug•', , 1 JI
```

S· rdenamos esta mabiz, ¿quf orden se aplicar6 finalmente a los valores? Como sa~os qui contenidos representan. existen dos posibilidades al menos. Podemos ordenar los produc:tm en orden alfaWtico utilizando su descripci~ o hac:erl~nummcamente por su precio. Ambas opciones IOJ\posibles, pero necesitamos utilizar la función uaort ( ) y decirle a PHP cómo comparar los artfculos. Para ello tenemos que escribir nuestra propia función de comparación.

El siguiente código ordena esta mabiz en orden alfabftico utilizando la segunda columna de la mabiz: la descripción.

```
funcciOG cCl111)areCSx,  Syl
(
   if I Sx(ll  •• Sy|ll   I
      recura 0,
   el8e  if  ( $X(1)  e  $y(1)   J
      recura  -11
   el••
      recura  11


ú•orc(Sproducc•,   'c011pare•Il
```

Huta el momento, hemos utilizado varias funciones incorporadas en PHP. P~a ordenar esta matriz, hemos definido una función propia. En un capitulo posterior íeanalizar6 en detalle la creación de funciones.

-:¡!:~'·° : :: t~d~~~~~~~~-!~~¡~~=

funciones toman par6metros o argumentos. Nuestra función compare () toma dos, uno llamado x y otro llunado y. El objetivo de esta función es tomar dos valores y determinar su orden.

En este ejemplo, los par6metros x e y ser6n dos matrices dentro de la ma~ principal y cada una representan un producto. Para acceder a la descripción d~ la mabiz x, escribimos $x (1J porque se trata del segundo elemento de estas matrices y la numeración comienza en cero. Utilizamos $x[l) y $y(1] para comparar las descripciones de las mabices pasadas en la función.

Al final de la función, se puede devolver una respuesta al código de llamada. Para devolver un valor, utilizamos la palabra clave return. Por ejemplo, la lfnea return 1; deYuelve el valor 1 al código que invocó la función.

Para utilizar la función compare () con usort ( ), debemos comparar $x e $y. La función debe devolver o si $x es igual a $y, un número negativo si es menor y un número positi'fO si es mayor. La función devolver6 o, 1,o -1, según los valores de $xe $y.

La linea final del código llama·• la función incorporada de PHP uaort () ~entro de la matriz que queremos ordenar ($producu) y el nombre de la función de comparación (compare ()).

Si quisiframos ordenar la mabiz de otra forma, bastaría con escribir una fundón de comparación diferente. Para ordenar la matriz por el precio, debemos examinar la tercera columna de la mabiz y crear la siguiente fund6n de comparación

```
funccion eo11pare($x,  Sy)
(
U  ( $x(2J  •• $y(2]  J
   recura 0,
el•e if I Sx[2]  e $y[2J
   recura -11
el••
   recura 11
```

Al llamar a uaort ($producta,compare), la matriz se ordenar6 en orden ascendente por el precio.

La "u" de usort ( ) equivale a "usuario" porque requiere una función de comparación definicla por el usuario. Las versiones uaaort () y uksort ( ) de aaort y kaort también requieren una función definida por el usuario.

Al igual que aaort (›, la función uasort (› debería utilizarse al ordenar ~ matriz asociativa por su valor. Utilice asort si los valores son números o texto. Defina una función de comparación y utilice uaaort () si los valores IOJ\ objetos más c:oinplicadosque las mabices. Al igual que kaort (), uksort ( ) debería utilizarse para ordenar una mabiz asociativa por su clave. Utilice ~aort () si sus claves son números o texto. Defina una función de comparación y utilice ukaort ( ) si sus claves son objetos mú complicados que matrices.

Ordenacione.s de usuario inversas

sort (), asort () y ksort () tienen funciones inversas que se establecen colocando una "r" delante del nombre de la función. Las ordenaciones definidas por el usuario no disponen de variantes inversas, pero se puede ordenar una matriz multidimensional en orden inverso. Para ello, bastará con esc:ribiruna función de comparación que devuelva los valores al revé. Para ordenar una mabiz en orden inverso, la lundón tendr6 que devolver 1 si $x es menor que $y y - 1 si $x es mayor que $y. Por ejemplo:

```
funccioa rever•eCompare($x, $yJ
(
if  ( $x[21  •• $y(2J  J
   recura o:
elae if I Sx[2)  e  Sy[2J  J
   retura l;
ei.e
   recura  ·l:
```

La función uaort ($producta, ·reverseCompare·) dar6 como resultado una mabiz en orden descendente por el precio.

EL_HOLISTICO@HOTMAIL.COM - 2006

Reordenar matrices

En algunas aplicaciones puede que necesitemos manipular el orden de las ~trices de varias formas. La función shuffle (I reordena de manera aleatona los elementos de la matriz. La función array_reverse (I devuelve una copia de la matriz con todos los elementos en orden inverso.

Utilizar shuffle()

Bob quiere destacar un número reducido de sus productos en la página principal del sitio. El número de productos que comercializa es grande, pero quiere seleccionar tres productos de manera aleatoria para que aparezcan en la página principal. Su intención es que los visitantes que vuelvan al sitio vean algo diferente en cada visita y no se aburran. Resultaría muy sencillo conseguir este objetivo si todos los productos se incluyeran en una matriz. El lis~do 3.1 muestra ~ imágenes seleccionadas de manera aleatoria. Para ello, se aplica un orden aleatorio a los elementos de la matriz y se muestra el primero de los tres artículos.

Uatado 3.1. bobs_front_page.php. Uso de PHP para generar una p6gina principal dln6mica en el sitio Bob's Auto Parta.

```
c?pbp
   $p1cture•.  array('t1re.jpg',    •o11.jpg•,  'epark_plug.jpg',
                     •door.jpg•,   •eteerlng_wbeel.jpg•,
                     'tbeinaoatat.jpg•,  •w1per_blade.jpg',
                     lgáiik9t.jP9•;    'brakeJl&d~fpgrJ i

   •buffle($p1cturea)1
1..
cbt111,.
cbead ..
   ct1tle..llo!a'a Auto Partac/t1tle..
c/bead.
cbody..
   ccenter,
      chl>Bob'a Auto Partac/bl>
      eta.ble w1dtb • 1001..
         ctr>
c7pbp
   for ( $1 • 01  $1 e 31  $1++
   (
      -«:bo 'ctd al1gn-•center•..c1115l •re•••;
      ecbo $p1cturea($1r
      ecllo '•v1dtb••100•  be1gbt•"100°u/td"'1

1..
         c/tr..
      e/table..
   c/center.
```



```
c/body ..
c/bt111,.
```

Como el código selecciona imágenes aleatorias, genera una página diferente prá~am~te cada vez que se carga, como se muestra en la figura 3.5.

Figura 3.5. La función shuffle() nos permite mostrar tres productos seleccionados de manera aleatoria.

En versiones anteriores de PHP la función shuffle II requería que se incrementara el generador de números aleatorios mediante la invocación de srand(I. Este paso ya no es necesario.

La función shuffle (I no se distingue por su pasado. En las'."ersiones más antiguas de PHP, no mezclaba muy bien los resultados aleatoriamlbte. En la versión 4.2.x para Windows no funcionaba en absoluto-(devoMa el mismo resultado--inicial). En la versión 5, parece funcionar. Nuestro consejo es probarla en el servidor para comprobar sus resultados.

Como realmente no necesitamos la totalidad de la matriz reordenada, podemos conseguir el mismo resultado por medio de la función array _rand ( ).

Utilizar array _reverse()

La función array_reverse (I toma una matriz y crea una nueva invirtiendo eJ orden de los elementos. Por ejemplo, existen varias formas de crear una matriz que incluya una cuenta atrás de diez a ww.

Si utilizamos range (I únicamente se aeerá una secuencia ascendente. Por ello, debemos utilizar rsort ( ) para ordenar los números en orden descendente. Opcionalmente, podemos crear la matriz elemento a elemento escribiendo un bucle for:

```
Snumber• • array();
for($1•101  $1..01  $1--1
   array_puab( $nullbera,  $1 )1
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Los bucles for (I pueden tener un orden descendente com!)áte. Establecem~ un valor alto como valor inicial y al final de cada bucle utilizamos -- para ~uar el contador a uno.

Creamos una matriz vada y utiliDmos ar:ray _puah (I para que cada elemento agregue uno nuevo al final de una matriz. La función contraria de array _puah (I es array _pop ( ). Esta función elimina y devuelve un elemento desde el final de una matriz.

TambWn podemos utilizar la función array _revene (I para invertir la matriz creada por range ( ).

```
Snullber• • ra1199(1,1011
Snullber• • array_raverae(Snumber•I:
```

Recuerde que array _revene (I devuelve una copia modifi~da de lama~. Como no queremos la matriz original, almacenamos la nueva matriZ sobre el ongi-nal. Si los datos son simplemente un intervalo de enteros, puede crearlo en orden inverso si pasa -1 como par6metro opcional de range (l:

```
Sn\lllbera • range(l0, l, ·ll;
```

Cargar matrices desde archivos

En un capitulo anterior, almacenamos los pedidos de los clientes en un archivo. Cada linea del archivo presenta un aspecto parecido a áte:

```
~-is,,~. ·nt1 1 Ap"&"il" c-.:1:r..-i--or1-•-eperkplup -$"3t~ 00. 2:l SbGEt.-5t.~ s..llt~
```

Para procesar o servir este pedido, podemos volver a carprlo en una matriz. El listado 3.2 muestra el archivo actual de pedidos.

U.taclo 3.2. vtewo,de,s.php. U.O de PHP para mostrar loe pecllldoa de Bob.

```
c?pbp
//cr.. un nombre de variable c:or:o
SDOCDNSIIT_ROO!' • s_ssavsa[•DOCUIIDT_R001")      I

Sordera• fila(•$DOCDIISIIT_ROO"r/ •• /ordar•/ordare.txt•)1

$numbar of ordar• • count($ordaral;
if (Snumber_of_ordara •• 01
(
   ec:bo ·cp>catZWlg>IIO ordara pmlding.
         Pleaa• try again later.c/atrong>c/p>'I
)
for ($i•01 $1c$nullber_of_ordera, $1++1
{
   ec:llo $ordar•[$11 • ' cbr />' 1
```


Esta secuencia de comandos genera pncticamente el mismo resultado que el stado 2.2 del capítulo anterior, que se muestra en la figura 2.4. Esta vez vamos a utilizar la función file () que carga el archivo entero en una matriz. Cada línea del archivo se convierte en un elemento de una matriz.

Este código tambim utiliza la funci6n count () para comprobar el número de elementos que contiene una matriz.

Adicionalmente, podríamos cargar cada sección del pedido en elementos de matriz separados para procesar las secciones de manera independiente o para aplicarles un formato mú llamativo. Eso es lo que hace el listado 3.3 exactamente.

Listado 3.3. vleworders2.php. Uso de PHP para eeparar, aplicar fonnato y mostrar loa pedidos de Bob.

```
c?php
   //cree un nombre da variable corto
   $DOCIIIIBIIT_ROOT • $_SUVBR(' DOCUNDIT_ROOT' I:
?>
chtal>
chead>
   ctitle>Bob•a Auto Parta - CUat.0ller Ordarac/title>
c/head>
cbody>
chl>lab'• Auto Partac/111>
cll2>CVato.ar Ordarac/h2>
c?php
   //IAa el archi1'1D 0011Pleto.
   //e.da pedido ae coa,rierte ea un ale..nto de la ..tris
   Sordera• fileC°SDOCmlSIIT_ROOT/.•/ordara/ordara.txt•I:
   // c:uante el na.ero de pedido• da la ..tris
   $llwiliir ot"ordira"7coünI(feiiiler•lt"
   if ($number_of_ordara •• 0)
   (
      echo •cp>catroag>llo ordara panding.
         Plaaae try ~in later.c/atroag>c/p>'1

   echo •ctable hordar•l>\n•;
   echo •ctr>cth bgc:olor••tc:ccc:FP°>Ordar Datac/th>
         cth bgcolor•"lccc:c:FP°>Tire•c/th>
         cth bgcolor•"ICCCCPP">Oilc/th>
         cth bgcolor••tc:cc:c:n•>Spark Plugac/tll>
         cth bgcolor•"lccc:c:FP">Totalc/th>
         cth bgc:olor•"lccc:c:FP°>Addreaac/th>
      ctr>';
   for (Si•Or $ic$DVllber_of_ordara1 $1++1
   (
      //divida c:ada llnea
      Slin• • axplode( •\t•, Sordera[Sil 11
      // Mantenga ordenado Gnic...nc• al no.ero da art1cv.loa
      Slina(ll • intval( Sline[l) 11
      $Une(2J • intval( $Une (21 11
      $11ne(JJ • intval( Sline(JI I;
      // --tre cada pedido
      echo •ctr>ctd>Sline(Olc/td>
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
            ctd align••ri9bt">$liae(llc/td>
            ctd ali911••ri9bt">$liael2lc/td>
            ctd aliga•"rigbt">$lina(3)c/td>
            ctd align••rigbt">$lina(4lc/td>
            ctd>$lina15Jc/td>
        c/tr>•;
    |
    acilo •e/tabla>";
?>
c/bocly>
c/ht•l>
```

El código del listado 3.3 carga el archivo co~pleto ~ una matriz pero a d~da del ejemplo del listado 3.2, aqui utilizamos la función e~lod~(1 para d~Vldir cada linea para poder procesar y aplicar formato antes de lD1prest6n. En la figura 3.6 se ilustra el resultado de esta secuencia de comandos.

Figura 3.8. Tras dividir los reglsfrosdel pedido con la función explode, podemos colocar cada Hc:c:ión del pedido en una celda de tabla diferente para mejorar su aspecto.

La función explode tiene la siguiente sintaxis:

```
array axplodll(atring aeparador, atring cadeaa (, int lf•1tall
```

En ~l capitulo anterior, utilizamos el car6cter de tabulación como delimitador al almacena~estos datos. Por ello, aqui llamamos a:

```
axplodllC •\t•,  $ordaral$1)  1
```

Este código divide las cadenas puada en partes. Cada carácter de tabulación separa los elementos. Por ejemplo, la cadena

```
•1s,42,2otb  April\t4 tirea\tl oil\tl apark pluga\t$434.00\t
22 Sbort St, S..lltown"
```

se divide en •1s:,2, 20th April•, •, tires•, •1 oil•, •& apark plugs•, •$,34.00• y •22 Short se , s-lltOWD•.

Ffjese en que el parámetro opcional *limite* se puede utilizar para restringir el número múimo de partes devueltas.

En este código no se ha realizado una gran cantidad de operaciones de procesamiento. En lugar de devolver los artículos en cada linea, sólo mostramos el número de cada uno y agregamos una fila de encabezados para mostrar quf representan.

Existen varias formas de extraer números de estas cadenas. En este caso, hemos utilizado la función intval (). Como se mencionó en un capitulo anterior, intval (1 convierte una cadena en un entero. El proceso de conversión resulta razonablemente inteligente e ignorará partes, como la etiqueta de este ejemplo, que no se pueden convertir en un entero. Eri el siguiente capitulo abordaremos varias **formas de procesar cadenas.**

Otras manipulaciones de matrices

Hasta el momento, sólo hemos analizado aproximadamente la mitad de las funciones disponibles para el procesamiento de matrices. Existen otras muchas ~ue resultarán de utilidad de vez en cuando.

Desplazarse c:.lentro de una matriz con eachQ, current(), reset(), end(), next(), pos() y prev(j

Ya mencionamos anteriormente que todas las matrices constan 41! un puntero intemo que apunta al elemento actual de la matriz. Ya se utilizó ante!'este puntero indixectanwutecon-la fanción each-0; pero podemos-utilizai-lo directamente-y-manipularlo. Si crea una nueva matriz, el puntero actual se inicializar6 con el puntero al primer elemento de la matriz. Si llamamos a current ($nombre_matriz) se devolverá la primer elemento.

Si llamamos a next () o a each (), el puntero avanzar6 un elemento. Si llamamos a each ($nombre_matriz), se devuelve el elemento actual antes de que el puntero avance. La función next (1 se comporta de manera ligeramente diferente: si llamamos a next ($nombre_matriz)• el puntero avanzar6 y devolver6 el nuevo elemento actual.

Ya hemos visto que reaet () devuelve el puntero al primer elemento de la matriz. De manera similar, si llamamos a end ($nombre_ -tri z)• el puntero se enviará al final de la matriz. reaet () y end (1 devuelven el primer y el último elemento de la matriz, respectivamente. Para recorrer una matriz en orden inverso, podemos utilizar end () y prev (). La función prev () es la opuesta a la función next (1. Mueve el puntero actual un puesto hacia atrás y devuelve el elemento actual. Por ejemplo, el siguiente código muestra ~ matriz en orden inverso:

```
$vall1• • ad  CSarray1
vbile  1$9aluel
(
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
echo o$valuecbr />ol
Svalue • prev(Sarrayll
```

Si $arra y se declarara de la siguiente forma:

```
sarray • array|l, 2, 31;
```

El resultado se mostraría en el navegador de la siguiente forma:

```
3
2
1
```

El uso de eac:h (), c:urrent O , reset (), end () , next O , pos () y prev () nos permite escribir nuestro propio código para desplazamos a través de una matriz en cualquier orden.

Aplicar una función a cada elemento de una matriz: array_walk()

En ocasiones, puede que necesitemos trabajar o modificar cada elemento de un matriz de la misma forma. Para ello, podemos utilizar la función array_valle. La sintaxis de esta función es la siguiente:

```
array_walk(array ..c. atring EWlc, [mixed datoa_de_uauar:1.ol l
```

~manera-similar-a-como-hicimos. mteriormente al llamar a usC)z:~ !1, 1~ fun~ción array walk ( ) espera que declaremos una función propia. Como puede apreciar array-walk () toma tres parámetros. El primero, *mt1t,* es la matriz que se pro~esará. ET segundo, *func,* es el nombre de una fundón definida por el usu~o que se aplicará a cada elemento de la matriz. El tercer parámetro, *d1ltos_de_1ls111no,* es opcimal. Si lo utiliza, se pasará a través de su fundón co~o pará~etro. ~n_un instante veremos cómo funciona. Podrlamos crear una funaón práctica definida por el usuario que muestre cada elemento con opdones de formato diferentes. El siguiente é6digo visualiza cada elemento en una línea llamando a la función definida por el usuario my_print (l con cada elemento de $arra y:

```
functioa ay_llrint(Svaluel
l
   echo 'svaluedlr />ol
}
array_valk1$array, 'my.Jlrint');
```

La función débe tener una firma especial. Para cada elemento de la matriz la función array_walk toma la clave y el valor almacenado en la matriz y todo lo que pase como *datos_de_usurio,* y llama a la función de la siguiente forma:

```
suiuaci6a(valor, clava, datoa_da_USDariol
```


En.la mayor parte de los casos, la función sólo utilizará los valores de la matriz. En algunos casos, puede que también necesite pasar un parámetro a la función utilizando el parámetro *datos_dt_usuario.* En ocasiones puede que esti interesado en la clave de cada elemento asf como en su valor. Como en el caso de MyPrint () la función puede optar por ignorar la clave y el parámetro *datos_de_usurio.*

Podemos escribir un ejemplo un poco más complicado, en el que una función modifique los valores de las matrices y necesite un parámetro. Pfjese en que a pesar de no estar interesad~ en la clave, tenemos que aceptarla para poder aceptar el tercer parámetro.

```
fuaction myMultiply|•Svalua, Skay, Sfactorl
{
   svalue •• $factor;
}
array_valk<•Sarray, •my_multiply•, 31l
```

En este ejemplo, se define una función,my_multiply() ,que multiplicará cada elemento de la matriz por un factor suministrado. Necesitamos utilizar el tercer parámetro opcional de arra y_~alk () para tomar un parámetro que pasar a nuestra función y utilizarlo como factor por el que multiplicar. Como necesitamos este parámetro, podemos definir la función my_multiply(} para que tome tres parámetros: un valor de elemento de matriz ($value), una clave de elemento de. matriz ($key) y otro parámetro ($factor). Vamos a optar por ignorar la clave.

Un punto sutil en el que fijarse es la forma de pasar $value. El símbolo & colocado delante del nombre de la variable en la definición de my mult iply () significa que $val ue se pasará por referencia. Este modo de pasar variables permite que la función modifique los contenidos de la matriz.

Analizaremos este modc;-eii un capítulo posterior: Por el-mómeñm;-ffjfte--sm=--- plemente en que al pasar por referencia se utiliza un símbolo & delante del nombre de la variable.

Contar elementos de una matriz: count(), sizeof() y array_count_values()

En un ejemplo anterior utilizamos la fundón count (l para contar la cantidad de elementos de una matriz de pedidos. La fundón aizeof (l tiene el mismo objetivo. Ambu funciones devuelven el número de elementos de una matriz que se le han pasado. Obtendremos 1 para el número de elemenlm en una variable escalar normal y 0 si pasamos una matriz vacía o una variable que no se haya establecido.

La función array_c:ount_valuea(l es más compleja. Si llama a array_count_values ($array), esta fundón contará el número de veces que tiene lugar cada valor único en la matriz $array. (Se determina por la l:ardinalidad de la matriz.) La función devuelve una matriz asociativa que contiene una tabla de frecuencia. Esta matriz contiene los valores exclusivos de $array como claves.

EL_HOLISTICO@HOTMAIL.COM - 2006

Cada clave tiene un valor numcfrico que indica las veces que tiene lugar la clave correspondiente en $array.

Por ejemplo, el siguiente código:

```
sa.rray.  arrayl•, 5, 1, 2, 3, 1, 2, 1);
sac.  array_count_valu••CSarrayl;
```

crea una matriz llamada $ac que contiene

Clave	Valor
4	1
5	1
1	3
2	2
3	1

Esto indica que 4, 5 y 3 aparecen una vez en $array, 1 aparece tres veces y 2 aparece dos veces.

Convertir matrices en variables escalares: extracto

Si tenemos una matriz indexada pero no numéricamente con una serie de pares de valor y clave, podemos convertirlos en un conjunto de variables escalares utilizando la función ext ractth·Esta función tiene la siguiente sintaxis:--

```
extractCarray matr1a_var [, int t1po_e.xtracci~nl  [. •tring preL1joJ 1:
```

El objetivo de extract (› es tomar una matriz y aear variables escalares con los nombres de las claves de la matriz. Los valores de estas variables se establecen en función de los valores de la matriz.

A continuaáón se incluye un ejemplo:

```
Sarray • array( 'k•y1•  •> 'valuel',  'key2'  •> •value2',  'key3'  •> •valuel');
extractl$arrayl
ecbo °Sk•yl Skey2 SkeyJ•;
```

Este código genera el siguiente resultado:

```
v&luel value2 valuel
```

La matriz tiene tres elementos con claves: keyl, key2 y key3. La función extract (› nos permite aear tres variables escalares, $keyl, $key2 y $key3. Por el resultlldo podemos ver que los valores de $keyl, $key2 y $key3 son ·valuel •, ·valuel2 · y ·value3 •,respectivamente.Estos valores proceden de Ja matriz original.

~ funáón extract () consta de dos parmnetros opáonalés: *tq,o_afNcd6n* y *prefijo.* La variable tipo_ extracción indica a extract () cómo resolver las colisiones. &tas tienen lugar cuando ya existe una variable con el mismo nombre que una clave. La respuesta predeterminada consiste en sobrescribirla variable eXistente. En la tabla 3.2 se recogen los valores posibles de este parámetro.

THIII 3.2. Valores permitidos del perémefro llpos_mdrllccl6n de la func:l6n extrec:i().

Tipo	Significado
DTR_OVBRlfRITB	SoInscribe la variable existente cuando llene lugar una coUslón.
DTR_$ltIP	Salte un elemento cuando tiene lugar una colisión.
BXTR_PRBFIX_SANB	e,.. una variable denominada $prefiJC_key cuando Uene lugar una colisión. Se debe incluir el penlmetro prefijo.
DTR_PRBFIX_ALL	Colocadelante de lados los nombres de variables el valor del prefijo. Se debe inclur el peréiMlro prefijo.
BXTR_PRBFIX_IJIVALID	Coloca prefijo delante de todos los nombras da variables que en caso contrario no serian *v@dos* (por ejemplo, nombres de vslables numéricos). Si debe incluir el parérnetro prefijo.
BXTR_IP_BXISTS	Extrae solamente variables que ya ·ii..n (es decir, rellena las variables existentes can ·.. procedentes de.la matriz). Esta opción se agregó a la versión 4.2.0 Y-rasulta úlll pera convertir, por ejemplo, $ RBQUBST en un conjunto de variables vélldas.
BXTR_PRBFIX_IF_BXISTS	Sólo cru una Ya'lión can prefi¡o si ya exillie9e una versión sin prefijo. Esta opción se agragó en la versión 4.2.0.
IXTR_RBFS	Extrae varlablH como referencia. Este opción se agregó en la versión 4.3.0.

Las opciones más útiles son las predeterminadas, EXTR_OVERWRITE, y EXTR_PREFIX_ALL. Las otras funciones pueden resultar de utilidad si sabemos ~e se va a producir una colisión y queremos saltarla o utilizar prefijos. A continuación se recoge un sencillo ejemplo utilizando EXTR_PREFIX_ALL. Como puede observar, las variables creadas se denominan prefix_guión bajo_nombre de clave.

```
$array · arráy( 'key1•  -> 'valuel',  •te)'2'  -> 'value2•,  'keyJ•  .,.  •valueJ•.k
eztract(S&rray, EXTll_PRBFIX_ALL, 'myPrefix'I;
ec:ba °$myPrefix_keyl SmyPr.rix_keyZ SmyPrefix_keyl•;
```

Este código devolverá de nuevo valuel vaiue:z value3.

EL_HOLISTICO@HOTMAIL.COM - 2006

Tenga en cuenta que para que extract () extraiga un elemento, la clave de dicho elemento debe ser un nombre válido de variable, lo que significa que se no se tendr6n en cuenta las claves que comiencen por números o que incluyan espacios.

Lecturas adicionales

En este capitulo se trata lo que en nuestra opinión son las funciones de matriz más útiles de PHP, pero no se han estudiado todas. En el manual en linea de PHP, disponible en bttp: //www.pbp. net /array, encontrar, una breve descripción de todas ellas.

A continuación

En el siguiente capitulo se analizann las funciones de procesamiento. Veremos c6~ realizar operaciones de búsqueda, sustitución, división y combinación de cadenas. Asimismo, se examinarút las potentes funciones de expresiones regulares capaces de realizar pr6cticamente cualquier operación sobre una cadena-

EL_HOLISTICO@HOTMAIL.COM - 2006

4

Manipular cadenas y expresiones regulares

En este capitulo, veremos cómo puede utilizar las funciones de cadena de PHP para aplicar formato y manipular texto. También veremos cómo utilizar funciones de cadena o funciones de expresiones regulares para buscar (y reemplazar) palabras, frases y otros patrones dentro de una cadena. Estas funciones resultan útiles en m':lc.bos contextos. Con frecuencia se suele limpiar o modificar el formato de los datos recibidos de los usuarios para almacenarlos en una base de datos. Las fundones de búsqueda son magnificas para crear motores de búsqueda (entre otras cosas). En este capitulo, analizaremos los siguientes aspectos:

- Formato de cadenas
- Combinar y dividir cadenas
- Comparar cadenas
- Coincidencia y sustitución de subcadenas con funciones de cadena
- Utilizar expresiones regulares

Crear una aplicación de ejemplo: Smart Form Mail

En este capítulo, vamos a examinar las funciones de cadena y de expresiones regulares en el contexto de la aplicación Smart Form Mail. Agregaremos estas se-.

EL_HOLISTICO@HOTMAIL.COM - 2006

cuenciu de commdos al sitio Bob's Autor Parts que hemos desarrollado en los últimos capítulos.

En concreto, vamos a crear un sencillo formulario para que los dientes de Bob expresen sus quejas y agradecimientos, como se ilustra en la figura 4.1. Sin embargo, nuestra aplicación incorporar, una mejora con respecto a los formularios de este tipo que se incluyen en la Web. En lugar de remitir el formulario por correo electrónico a una dirección genmca como informacióueej emplo. com, intentaremos que su procesamiento resulte mAsavanzado. Para ello, se analizar' el texto en busca de palabras clave y frases, y se remitirá el correo electrónico al empleado apropiado de la compai\la de Bob. Por ejemplo, si el correo contiene la palabra "advertising", el formulario podría dirigirse al departamentp de marketing. Si el correo procede de un cliente importante, se dirigid directamente a Bob.

Flgur. 4.1. El formulllrio para comentarios de Bob pregunta a loe dientes su nombre, dlrecdón da COffllO elec:tr6nlc:o y comenwrlae.

Comenzaremos por la sencilla secuencia de comandos que se muestra en el listado 4.1 e iremos agregando elementos progresivamente.

Llatado 4.1. proceafeedback.php. Secuencia de comandos baica de Email Forms Contenta.

```
c?pbp
   //cree llOllbre• de variable•
   S...-.•S_POST('n&11e');
   S...il•S_POST('e..il'I;
   Sfeedback•S-~l'feedback•J1

   $toaddress = 'informacion@ejemplo.com';
```



```
   Sllllbject • 'Feedback froa IMb •ite•;
   ...ilconteat •'CU.taller.,....,'.s ......•\n•
                  'CU.taller  ...11, '.se ..il.•\n•
                  .•cu•tomer  coament•'  \n°.$feedback.•\n•;
   $f~re••   •  ·rr-, -b•erver&ejetll(llo.com';

   119ill$to.ddre••·  S•ubject,  $119ilcootent, Sfromaddre••I;
?>
cbtml>
cbeaib
   ctitle>lob'• Auto Part• - Feedback Submittedc/title>
c/bead>
cbody>
cbl>Feedb•ck •ubmittedc/bl>
cp>Y-r  feedtwlck baa beea •eat.c/p>
c/body>
c/bt111>
```

Por regla general, debería comprobar si los usuarios han rellenado todos los campos obligatorios del formulario con ayuda de iHet (), por ejemplo. Hemos omitido esta operación en la secuencia de comandos u(como en otros ejemplos para no alargar el anAlisis.

En esta secuencia de comandos, hemos concatenado los campoidel formulario y hemos utilizado la función mail () de PHP para enviar un correo electr6nico a informaci6neejemplo.com. Es una dilección de corno electtónico ficticia. Si desea probar el código de este capítulo, sustitúyala por una dirección real. Como todavía IJO hemos utilizado mail (), a continuación analizaremos su funcionamiento. La función mail ( ) envía correos elec:bónicos.Su sintaxis es la siguiente:

```
bool ..ilC•tring .PU•, •tring ••llllco,  •triag —n••:l•,
     •trÜlf. (ucabHado•...•dJcionalH·[,  -«riD!t  -per ....  .ro•..:;N1cicmaJe•J  J h  ..-
```

Los primeros tres parAmetros son obligatorios y representan la dirección a la que dirigir el correo electrónico, la linea de asunto y los contenidos del mensaje, respectivamente. El cuarto par'metro se puede utilizar para enviar otros encabezados v61idos. Los encabezados v61idos de corno electrónicos se desaiben en el documento RFC822, que est, disponible en linea si desea obtener mAs detalles. (Los RFC o Solicitudes de comentarios son la fuente de una gran cantidad de estAndares de Internet, como se comentar' en un capítulo posterior.) Aquí hemos utilizado el cuarto parAmetro para agregar una c:lirec:dml • Prom: • al~ electr6nico. También puede utilizarlo para agregar campos •aeply-To: • y •ce:•, entre otros. Si desea induir mú de un encabezado, sólo ser' necesario separarlos mediante caracteres de ,nueva linea (\n) en la cadena, de la siguiente forma:

```
$additional_beeder••°Proa: veb8erYer&ejaaplo.coa'\n•
          .'Reply·To:  bobeejeaplo.c-·;
```

El quinto par'metro se puede utilizar para pasar otro argumento al programa que tenga configurado para enviu corno. Para poder utilizar la función email ( ), configure su instalación de PHP para que apunte al programa utilizado para enviar

EL_HOLISTICO@HOTMAIL.COM - 2006

correos electrónicos. Si la secuencia de comandos no funciona correctamente, consulte el apéndice A. A lo largo de este capítulo, optimizaremos esta secuencia de comandos básica utilizando funciones de procesamiento de cadenas y expresiones regulares de PHP.

Aplicar formato a cadenas

A menudo es necesario limpiar las cadenas que envían los usuarios (por regla general procedentes de una interfaz de formulario HrML) pa~ poder utilizarlas.

Limpiar cadenas: chop(), ltrim() y trim()

El primer paso del proceso de limpieza consiste en quitar todos los espacios en blanco no necesarios de la cadena. Aunque no resulta obligatorio, puede resultar útil si se va a almacenar la cadena en un archivo o base de datos, o si vamos a compararla con otras cadenas.

PHP incorpora tres funciones que resultan de utilidad en este sentido. Utilizaremos la función trim () para limpiar los datos de entrada de la siguiente forma:

```
$n&1Mt•tria($aa.el,
se1111il•tria($e ..111,
$feedback•tria($feedback11
```

La función trim (› elimina los espacios en blanco desde el principio al final de una cadena, y devuelve la cadena resultante. De manera predeterminada, limpia lo&caracteres-de nueva linea y retorno del carm (\n.y \ r),. tabulador!!$ ~~ontales y verticales (\t y \xOB), caracteres de final de cadena (\O) y espacios. También permite pasar un segundo parllmetro para incorporar otros caracteres que eliminar a la lista predeterminada. En función de lo que desee, puede que le interese utilizar las funciones ltrim() o rtrim(). Ambas son similares a tri111(): toman la cadena en cuestión como parúnetro y devuelven la cadena con formato. La diferencia entre estas tres es que trim (› elimina los espacios en blanco desde el principio al final de la cadena, ltrim() elimina los espacios en blanco desde el principio (o desde la izquierda) únicamente y rtrim ‹) elimina los espacios en blanco desde el final (o desde la derecha) únicamente.

Aplicar formato a cadenas para presentaciones

PHP consta de un conjunto de funciones que permiten volver a aplicar formato a una cadena de diferentes formas.

Utiliz.# formato. HTML: la función nl2br()

La función ti12br () toma una cadena como párúnetro y sustituye todas las lineas nuevas con la etiqueta XHTML <br / > (o la etiqueta cbr> en las versiones

anteriores a la 4.0.S). Esta función resulta útil para imprimir una cadena • grm. tamaño en el navegador. Por ejemplo, hemos utilizado esta función para aplic:n formato a la información recibida del cliente y devolverla:

```
cp>Your feedback 1•hown below1 ha• been •ent.c/p>
cp>c7 echo nl2br($..ilcontent1, ?> c/p>
```

Recuerde que tITML no tiene en cuenta los espacios en blanco sin procesar, por lo que si no filtramos este resultado a travá de nl2br (›, aparecerlln en una sobi línea (a excepción de las lineas nuevas forzadas por la ventana del navegador), como se ilustra en la figura 4.2.

Flgu111 4.2. El uso de la función nl2br() mejora la representación de cadenas largas dentro de HTML.

Aplicar formato a una cadena para su impresión

Hasta el momento, hemos utilizado la instrucción echo para mostrar caderiás en el navegador. PHP también incluye la instrucción print (), que realiza la misma función que echo, pero devuelve un valor (true o false, para denotar el resultado de la operación).

Estas dos técnicas imprimen una cadena sin mú. Puede aplicar formato más sofisticado utilizando las funciones printf (› y sprintf (). Estas funcionan básicamente de la misma forma, con la excepción de que la primera imprime una cadena con formato en el navegador y la segunda devuelve una cadena con formato. Si ha programado previamente en C, observará que estas funciones son iguales que sus versiones de C. Si no lo ha hecho, lleva cierto tiempo acostumbrarse pero resultan útiles y potentes. La sintaxis de estas funciones es la siguiente:

```
atring aprintf Catri119 ro.1'llato [, aixed arga•••JI
void printf Cetrin9 r0.1'llato (, mixed arga•••JJ
```

El primer parámetro pasado a ambas funciones es una cadena de formato que desaibe la forma básica del resultado con código de formato en lugar de variables.

EL_HOLISTICO@HOTMAIL.COM - 2006

Los deJMS parúnetre>S son variables que se sustituinn en la cadena de formato. Por ejemplo, con echo, utilizamos las variables deseadas para imprimir en linea. Por ejemplo:

```
echo •Total all0llllt of order 1• $total.•,
```

Para obtener el mismo resultado con printf (), se utilizar4:

```
pr1ntf (•Total amount of ordar 1• la.•, $total);
```

La secuencia •• utilizada en la cadena de formato se denomina especificación de conversión. Su objetivo es "ser reemplazada por una cade¡ia". En este caso, ser, sustituida con $total, interpretada como una cadena. Si el valor almacenado en $total fuera 12.4,ambu instrucciones imprimirian 12 . 4.

La ventaja de printf (> es que podelllOS utilizar una especificación de conversión mú útil para especificar que $total es un número de coma flotante y que deberla constar de dos decimales tras el punto, como se indica a continuación:

```
printf (•Total a80UDt of order 1• l.2f", $total);
```

La cadena de formato puede incluir variu especificaciones de conversión. Si tenemos o especificaciones de conv!!rsión, tendr4 n argumentos tras la cadena de formato. Cada especificación de conversión ser4 reemplazada por un argumento con nuevo formato en el orden en el que se enumere. Por ejemplo:

```
prinU (•Total -at of order ia.1.Zf lw1tla ahippi!MJ l.2fl •,
      $total, $total_ah1pping);
```

Las espeiifi.cadones de conversión utilizañn la variable $total pe'ro en la seguda se utillzar lavaria&le $t otil:_stiípping;

Cada especificación de conversión sigue el"mismo formato, a saber:

```
U• car•c:ter_de_nll•-1 1·J (aadluraJ l.pnc:ieidal tipo
```

Todas las especificaciones de conversión comienzan por un símbolo 1. Si desea imprimir un símbolo 1 necesitar, utilizar H.

a mrácter de „11„0 es opcional. Se utilizar4 para rellenar los espacios de la variable con la anchura especificada. Por ejemplo, podemos agregar ceros iniciales a un número en un contador. El car4cter de relleno predeterminado es el espacio. Si especifica un espacio o cero, no tendr, que atladir un apóstrofe (') como prefijo. Para cualquier otro car¸cter de relleno sí tendr4 que atladirlo.

El símbolo - es opcional. Especifica que los datos del campo se justificañn a la izquierda, en lugar de a la derecha (opción predeterminada).

El especific:ador and1un1 indica a printf () el espacio (en caracteres) que se debe dejar para que la variable se sustituya.

El esp¡ecificador prtd,i6n debería.comenzar por una coma decimal. Deberla contener el número de espacios deseados tras la coma decimal.

La parte final de la especificación es un código de tipo. En la tabla 4.1 se recoge un resumen.

Tabla 4.1. C6cllgoa de tipo de eapeclflcacl6n da conversl6n.

Tipo	Significado
b	
e	Se Interpreta como un entero y H Imprime como un caricter.
d	Se interpreta como un entero y • imprima como un número decimal.
f	Se interpreta como un doble y H imprime como un número con coma ftotante.
o	Se Interpreta como un ent9ro y H Imprime como un número octal.
8	Se interpreta como una cadena y H Imprime como una cadena.
u	Se interpreta como un entero y H Imprime como decimal lin signo.
X	Se interpreta como un entero y sa Imprime como un oomero decimal con minúsculas para loa dl9itoa a-f.
X	Se Interpreta como un entero y H Imprime como un número hexadecimal con mayúlcula para loa dlgilol A-F-.

. ——e la versión 4.0.6 se puede utilizar la numeración de argumentos, lo que • s1gmficaque no es necesario utilizar el mismo orden que las especificaciones de conversión. Por ejemplo:

```
printf (•Total &II0Ull t of order ia 12\$.2f (with ahippillg 11\$.2fl •,
     Stotal_ahipping, $total);
```

- ~~ éon agregiír-iá posición del argumento~reciameii~iraseTsfiñboio_t. seguido por el c:adcter de escape $. En este caso, 2\$ significa "sustituir con el segundo argumento de la lista". Este mitodo tambim se puede utilizar para repetir argumen~.

~xisten dos.áltemativ~ de estas funciones; vprintf () y vsprintf (). Estas vanantes admiten dos padmetros: la cadena de forma y una matriz de los argumen~ en lugar de un número variable de parúnetroe.

Cambiar mayúsculas y minúsculas en una cadena

Podemos modificar el uso de maymculu y minúsculas en una cadena. Este rec:uno ~ resulta especialmente 11til en nuestra aplicaci6n, pero examinaremos algunos e,emplos.

Si comenzamos por la cadena del asunto, $subject, que vamos a utilizar en nuestro correo electrónico, podemos cambiar el formato de mayúsculas y minúsculas con ayuda de variu fundones. El efecto de estas funciones se reswne-en la tabla 4.2. La primera columna muestra el nombre de la función, la segunda describe su'; electo, la tercera muestra cómo se aplicar4 a la cadena $subj ect y la última indicia qué valor se devotwr, de la función.

EL_HOLISTICO@HOTMAIL.COM - 2006

Tabl• 4.2. Func:lane1 .,.,. aplar mmyúlcula y mlníiscul•• • c:11den•1 y sus efectos.

Función	Descripción	Uso	Valor
		$subject	Información procedente del sitio Web INFORMACIÓN PROCEDENTE DEL SITIO WEB
scrcolower 11	c-ierte lac:adena enminúlculaa	scrcolower($subjeccJ	informaei6nprocedenledel IiIioWeb
ucfirat:11	Poneenmayúec:11L el primer caricterde la cadelle .. " ... c.,Kllr~	ucfirst: lSaubjecc l	• 1nfannacl0procadenladel sitio Web
ucwords()	Poneen-,üac:ula la primera letra de cada palabra de lll cadena que comience por un c:aric:ter alfab6llco	ucvords l$subj ec el	lnJarmacl6nProcedellle Del Sitio Web

Aplicar formato a cadenas para su almacenamiento: addslashes() y stripslashes()

.Aclemá$J~e ~tili?f. func;i°"'5 d~ cadffl.8 para mo~ifica~ el ~~to ~e~ ~de-nas visualmente, podemos utilizar parte de estas .funciones para variar el formato de cadenas para su almacenamiento en una base de datos. Aunque el tema de la escritura de cadenas en bases de datos no se examinará huta la segunda parte de este libro, seguidamente analizaremos cómo aplicar formato a cadenas para su almacenamiento en bases de datos.

Determinados caracteres son perfectamente válidos como parte de una cadena pero pueden originar problemas, en especial al insertar datos en una base de datos ya que éta podria interpretarlos como caracteres de control. Los caracteres problemáticos son lu comillu (simples o dobles), lo barras invertidas y el carácter NOL. Debemos buscar una forma de marcar estos caracteres para indicar a las bases de datos, como MySQL, que se trata de caracteres especiales y no de una secuencia de control.

Pan marcar estos caracteres como caracteres especiales, se agrega una barra invertida delante de ellos. Por ejemplo, las comillas dobles (•) se convertirán en \• (barra invertida ~da de lu comillas dobles) y la barra invertida (\) se convierte en \\'(doble barra invertida). (Esta regla se aplica universalmente a los caracteres especiales, de manera que si tiene \\ en una cadena, deberá sustituirla por \\\\.) PHP incorpora dos funciones específicamente dise1ladas para marcar caracteres

especiales. Antes de escribir cadenas en una base de datos, deberfa modificu III formato con la función addalaahea (), por ejemplo:

$feedbeck • addalaabaa($feedbac:kl;

Como muchas otras funciones de cadena, addalaahea () toma una cadeNl como parámetro y devuelve la cadena con formato nuevo.

La figura 4.3 muestra los efectos de utilizar estas fundones en la cadena.

Figura 4.3. Tras llamar a la función addslashel(), se coloc:anlln barra, delante de ladas las comilllls. La función slripslashea() quitará lal ba,..,_

Puede probar estas funciones en su servidor y obtener un resultado similar al ilustrado-en la figura-4-:4;-:-

FIgul'll 4.4. Todas lol c:11n1ctere1 conflictivos se han 81C11pmdo dol veces, lo que lignlftc:a que se han hablltado las cominas m6glca1.

EL_HOLISTICO@HOTMAIL.COM - 2006

Si puede ver el resultado, significa que su versión de PHP se ha configurado para ai\adir y eliminar barras de forma autom6tica. Esta posibilidad se controla por medio de la directiva de configuración magic_quotes_gpc, que sólo aparece habilitada de forma predeterminada en las nuevas versiones de PHP. Las letras gpc equivalen a GET, POST y cookie, lo que significa que a las variables que provengan de dichos origenes se les ailade comillas automáticamente. Puede comprobar si esta directiva está habilitada en su sistema con ayuda de la función get_magic_quotes_gpc (l, que devuelve true en caso afirmativo. Si su sistema cuenta con esta directiva, tendrA que invocar stripslasbes (l antes de mostrar datos del usuario; en caso contrario, se mostrar6n las barras.

El uso de comillas mágicas le permite escribir código mais port6til, como veremos en un capitulo posterior.

Combinar y dividir cadenas con funciones de cadena

A menudo, necesitamos examinar partes de una cadena por separado. Por ejemplo, puede que necesitemos examinar las palabras de una frase (para comprobar su ortografía, por ejemplo) o dividir un nombre o una dirección de correo electrónico en sus componentes. PHP incorpora varias funciones de cadena (y una función de expresión regular) que nos permite realizar esta tarea.

En nuestro ejemplo, Bob quiere recibir todos los comentarios procedentes de bigcustomer. com directamente, por lo que dividiremos la dirección de correo electrónica escrita por el.:cl~~te en p_a~ P-lml de~mu.nauI.se trata de clientes. importantes de Bob.

Utilizar explode(), implode() y join()

La primera función que podriamos utilizar para este objetivo es explode (l cuya sintaxis es la siguiente:

```
array explode(etrillg ••~redor, etriag' eatrade (, iDt ll•itell;
```

Esta función tom:a una entrada de cadena y la divide en partes en una cadena de separador especificada. Las partes se devuelven en una matriz. Puede limitar el número de partes con el parAmetro opcional de limitación, agregado en PHP 4.0.l.

Para obtener un nombre de dominio de la dirección de correo electrónico del cliente en su secuencia de comandos, podemos utilizar el siguiente código:

```
s—1 1_array • explode1 •••, s — 111 ,
```

Esta llamada a la función explode () divide la dirección de correo electrónico del cliente en dos partes: el nombre de usuario, que se puede almacenar en

$email_array [O) y el nombre de dominio en $email_array [1). Ahora ya podemos probar el nombre de dominio para determinar el origen del cliente y enviar sus comentarios a la persona apropiada.

```
if ($email_array(ll••'bi9cuet011er.c011')
   $toaddr••• • 'bobeejemplo.c011''l
elee
   $toaddreee • 'inforaaci6Deejemplo.c011";
```

Si el dominio estuviera en mayúsculas, la operación no funcionaria. Podemos evitar este problema convirtiendo el dominio a todo mayúsculas o todo minúsculas y realizar la comprobación seguidamente:

```
$email_erray(1) • etrtoupper (S•••il_array[l)I;
```

Puede invertir los efectos de explode (l utilizando las funciones implode (l o j oin (), que son idénticas. Por ejemplo:

```
$nev_email • implode('e', $.. il_erreyl,
```

Este código toma los elementos de la matriz $email_array y los combina con la cadena pasada en el primer parámetro. La llamada a la función es muy similar a explode (), pero el ~o es el contrario.

Utilizar strtok()

A diferencia de explode (), que divide una cadena en pequeñas partes de una sola vez, strtok (l va extrayendo partes (llamadas sfmbolos) de la cadena una a una. Esta función es una alternativa útil a la función explode para pn,cesar una cadena palabra a palabra.

La sintaxis de la propiedad strtok (l es la siguiente:

```
striag etrtok(atrin9 entrada, etring ••~redor);
```

El separador puede ser un carácter o una cadena de caracteres pero la cadena de entrada se dividirá en cada uno de los caracteres de la cadena del separador en lugar de en toda la cadena (como ocurría con explode).

La tarea de llamar a la función strtok (l no resulta tan sencilla como pudiera parecer por su sintaxis.

Para obtener el primer sfmbolo de una ca—, se invoca atrtok (l con la cadena divida en las partes deseadas y un separador. Para obtener las siguientes partes de la cadena, se pasa un único parámetro: el separador. La función mantiene su propio puntero interno dentro de la cadena. Si desea restablecer el puntero, puede pasarle la cadena de nuevo.

La función atrtok (l se suele utilizar de la siguiente forma:

```
$token• etrtok(Sfeedback, ' 'l;
ecbo $token.'cbr />';
vbile ($token!•'') .
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
    $toJc•n  •  •trtoltl'    •1,
    ectla  $toJc•n.'cbr    />';
},
```

Como de costumbre, conviene comprobar que el cliente ha escrito algún comentario en el formulario, utilizando la función empty () . En nuestro caso se ha omitido para no alargar el ejemplo.

Este código imprime cada símbolo del comentario del cliente en una linea distinta *y* recorre en bucle el comentario hasta que no quedan m4s símbolos. En las versiones de PHP anteriores a la 4.1.0, strtok <) no funcionaba de forma exactamente igual a como lo hace en C. Si una cadena de destino contiene dos instancias de un separador seguidas (en este caso dos espacios), strtok <> devuelve una cadena vaáa. Este resultado no se puede diferenciar de la cadena vacía que se devuelve al llegar al final de la cadena de destino. Así mismo, si uno de los símbolos es 0, se devolverá la cadena vacía. Esto convierte a la función strtok () de PHP en algo menos útil que la de C. La nueva versión funciona correctamente ya que ignora las cadenas vacías.

Utilizar substr()

t,a función su.bstr () permite acceder a una subcadena situada entre el punto iniéial *y* el punto final de una cadena. En nuestro ejemplo no resulta apropiado pero puede ser útil si intenta acceder a partes de cadenas con formato fijo.

La función su.bstr () tiene la siguiente sintaxis:

```
9tri1!9  .ub•trC•t!iD9.c11du11._int..in.ic.iol,_int.long.itudJ_); _
```

Esta función devuelve una subcadena copiada de *cadena*.

Los siguientes ejemplos utilizan esta cadena de prueba:

```
$t••t • 'Your cu•toaer ••rvice i• •xc•llent',
```

Si se llama a esta función con un número positivo desde el parámetro *inicio* (únicamente), se obtendrá la cadena desde la posición *inicio* hasta el final de la cadena. Por ejemplo:

```
eub•trCSteet, 11;
```

devuelve our customer service is excellent. Tenga en cuenta que la posidón inicial es 0, como ocurre con las matrices.

Si llama a la función su.bstr () con un valor *inicio* negativo (únicamente), obtendrá la cadena desde su final menos los caracteres *inicio* hasta el final de la cadena. Por ejempl~:

```
•ub•trC$tl111t,    ·9!;
```

En este caso se devuelve excellent.

El parámetro de longitud se puede utilizar para especificar un número deancteres que devolver (si es positivo) o el carácter final de la secuencia de retomo (li • negativo). Por ejemplo:

```
•ub•trC$te•t, o, ti,
```

devuelve los cuatro primeros caracteres de la cadena, en concreto, Your•. El siguiente código:

```
ecllo  •ub•tr1$te•t, t,  -13);
```

devuelve los caracteres situados entre el cuarto y el decimotercero, es decir, customer service. El primer carácter se encuentra en la ubicación 0, por lo que la ubicación 5 se corresponde al sexto carácter.

Comparar cadenas

Hasta el ~omento sólo hemos utilizado el operador ... para co~arar si dos cadenas son iguales. Pero PHP permite realizar comparaciones un poco más avanzadas. Las hemos dividido en dos categorías: coincidencias parda~ y otras. En primer lugar estudiaremos las demás y seguidamente pasaremos a estudiar las coincidencias parciales, para lo cual avanzaremos en el desarrollo del éj:emplo Smart Form.

Ordenar cadenas: strcmp(), strcasecmpQ y strnatcmpu

Estas funciones se pueden utilizar para ordenar cadenas y, por tanto, para ordenardatos.

La sintaxis de la función strcmp () es la siguiente:

```
int strc,op(•tri:ig c•dl, etring c•dZI:
```

La función espera recibir dos cadenas, que comparará. Si son iguales, devolverá o. Si la cadena cadl viene tras (o es mayor que) cad2 en orden lexicográfico, strcmp () devolverá un número mayor que cero. Si cadl es menor que cad2, strcmp () devolverá un número menor que cero. Esta función discrimina entre mayúsculas y minúsculas. La fundón strcasecmp (> es idmtica con la diferencia de que no discrimina entre mayúsculas y minúsculas.

La función •trcasecmp () y su pareja strnatcasecmp (), que no discrimina entre mayúsculas y minúsculas, se agregaron en PHP 4. Estas funciones comparan cadenas en función de un "orden natural", que se parece más a la forma en la que lo hace un humano. Por ejemplo, strcmp () ordenará la cadena •2• como mayor que la cadena •12• porque lexicográficamente iesulta mayor.

EL_HOLISTICO@HOTMAIL.COM - 2006

etrcaeecmp () hace exactamente lo contrario. Si desea saber más sobre el orden natural, dirijase a bttp: //www.naturalordersort, org.

Comprobar la longitud de una cadena con strlen()

Podemos comprobar la longitud de una cadena con la función etrlen ().Sise pasa en una cadena, devolverá su longitud. Por ejemplo, etrlen ('bello') devuelve 5.

Esta función se puede utilizar para validar datos de entrada. Considere la dirección de correo electrónico de nuestro formulario, almacenada en $email. Una forma elemental de validar una dirección de correo electrónico almacenada ~ $e—il consiste en comprobar su longitud. Si partimos de ~os álculos.aproximados, la longitud mínima de una dirección de correo electróruco_~ de SCa_S ~racteres, por ejemplo aea. to (suponiendo ~ue el códi~o de! país no ~ne doDIIIUOS de ~o nivel, y que el nombre del semdor y la direcoón sól~tienen ~ letra respectivamente). Por lo tanto, si la dirección no tiene esta longitud mínima, se generará un error:

```
if !•trlen($...il) e 6)
(
   eche> •Tbat e-il addre•• ie not valid'/
   éxit, ¡¡ finalice la ejecucien de la secuencia de c011&Ddo• de PHP
```

Obviamente, se trata de una forma muy simple de validar la información. En la siguie~te sección examinaremos formas mejores.

Buscar subcadenas y reemplazarlas con funciones de cadena

Resulta habitual comprobar si una subcadena dada está presente en una cadena de mayor tamaño. Esta coincidencia pardal suele ser más útil que comprobar la igualdad entre cadenas.

En nuestro ejemplo Smart Form, queremos buscar determinadas frases clave dentro de los comentarios de los clientes y enviar el correo al departamento adecuado. Si queremos enviar correos electrónicos relacionados con las tiendas de Bob al encargado del departamentos de ventas, tendremos que saber si aparece la palabra ·shop· (o alguna variación) en el mensaje.

Podríamos utilizar funciones ya vistas, como explode () o etrtok () para recuperai: palabras individuales del mensaje y compararlas utilizando el operador •• o etrcmp().

Sin embargo, podemos hacer lo mismo con una única llamada de función a una de las funciones de comparación de cadenas o de expresiones regulares. &tas se

utilizan para buscar un patrón dentro de una cadena. Examinaremos cada conjunto de funciones por separado.

Buscar cadenas en cadenas: strstr(), strchr(), strrchr() y stristr()

Para buscar una cadena dentro de una cadena, podemos utilizar cualquiera de las funciones strstr (), strchr (), strrchr () o stristr ().

La función strstr () es la más gen¸rica y se puede utilizar para buscar una cadena o carácter dentro de una cadena de mayor tamaño. Tenga en cuenta que en PHP, la función strcbr () es exactamente igual que la función strstr (), aunque su nombre implique que se utiliza para buscar un carácter en una cadena, de manera similar a la versión de C de esta función. En PHP, se pueden utilizar las dos funciones para buscar una cadena dentro de otra cadena, incluyendo la posiblidad de buscar una cadena que contenga un solo carácter.

La sintaxis de stretr () es la siguiente:

```
string strstr(striag pajar, string agujal;
```

En la función se pasa eJ parámetropwrjor sobre el que buscar y un parámetroopjo que encontrar. Si se encuentra una coincidencia exacta *deogujo,* la función devolverá el *pajar* desde la *agujo* en adelante o, de lo contrario, devolverá false. Si la *oga,jo* tiene lugar en más de una ocasión, la cadena devuelta comenzará desde'la primera ocurrencia de *oguja.*

Por ejemplo, en la aplicación Smart Form, podemos decidir dónde enviar el correo-electrónico dela siguiente forma=

```
$toaddrees • 'feedbackeejemplo.c-" /  // el valor predeterminado

// Cambie $toaddress si .. cumple el criterio
if (strstr($feedback, •sbop'))
   $toaddre•• • •retaileej911Plo.c-••,
else if (strstrl$feedback, •delivery'II
   $toaddress • 'fulfilaenteejemplo.com••,
else if (strstr($feedback, •bill'II
      $toaddre•• • 'accounte..jemplo.com••,
```

Este código busca determinadas palabras clave en el comentario del usuario y envía el correo a la persona adecuada. Por ejemplo, si el cliente escribe "I still haven't received delivery of my last order" (todavfa no he recibido mi último pedido), se detectará la cadena ·delivery" (entrega) y el comentario se enviará a fulf ilment•ej emplo. com. Existen dos variantes de strstr ( ). La primera es etristr (), que es prácticamente idfntica con la salvedad de que J!0 distingue entre mayúsculas y minúsculas. Esta función resulta útil en nuestra aplicación ya ;.. que el cliente podría escribir ·deli very• ,! Delivery• o ·DBLIVBRY•.

La segunda variante es strrchr (), que es, de nuevo, prácticamente icUntica, pero devuelve *pajor* desde la última ocurrencia de la *ogujo* en adelante.

EL_HOLISTICO@HOTMAIL.COM - 2006

Buscar la posición de una subcadena: strpos(), strrpos()

Las funciones strpos () y strrpo• () funcionan de la misma forma que strstr (), con la diferencia de que en lugar de devolver una subcadena, d~el-ven la posición numérica de una *11guja* en un *paj11r.* El manual de PHP recoauenda utilizar strpo• (› en lugar de strstr (› para comprobar la presencia de una cadena dentro de otra cadena, ya que se ejecuta: más rápidamente.

ta función strpo• ‹› tiene la siguiente sintaxis:

```
int at:poa(atrill9 JHi•r, atring •flUi•, int lde•pl••••i•DfOI l;
```

El valor entero devuelto representa la posición de la primera instancia de *11guji1* dentro de *paj11r.* El primer carácter es la posición 0 como de costumbre.

Por ejemplo, el siguiente código imprimirá el valor 4 en el navegador:

```
$te•t.  'Mello world';
echo atrpoe(Steet,  •o•I:
```

En este caso, sólo hemos pasado un carácter como *11guja,* pero puede ser una cadena de cualquier longitud.

El parámetro opcional de desplazamiento se utiliza para especificar un punto dentro de *pajar* a partir del cual iniciar la búsqueda. Por ejemplo:

```
echo atrpoe(Steat,  'o•,  51;
```

Este código imprimirá 7 en el navegador porque P!fi' ha comenza~~ a buscar el carácter *o* en la p~ición S y, por lo tanto, no verá el situado en la pos1aón 4...

t;a función strrpos-( ) es prácticamente idéntica, pero no devuelve !a ~1c~~ de la última instancia *duguji1enpaj11r.* Encualquiera de estos casos, *si11guji1* no está en la cadena, strpos () o strrpos () devolverán false, lo cual puede plantear un problema porque false en un lenguaje con control d6bil d~ tipos como PHP equivale a 0, es decir, al primer carácter de una cadena. Para evitar este problema podemos utilizar el operador ••• y probar los valores devueltos:

```
Sreeult • etrpoe($teat,  'H'I:
if  !Sreeult  ••• faleel
   echo 'llot found'
elae
   echo •Pound at poeition 0';
```

Tenga en cuenta que sólo funcionará en PHP 4 ya que en las versi~es ~teriores puede probar si es false examinando el valor devuelto par~detenrunar s1 se trata de una cadena {es decir, false).

Sustituir sÚbcadenas: str_replace() y substr_replace()

La función de buscar y reemplazar cadenas puede resultar extremadamente útil con cadenas. Se puede utilizar para personalizar documentos generados por PHP

(por e;emplo, sustitu~~ «nombre» con el nombre de una persona y «direcc1~n» con su. dirección). Tambi6n puede utilizarla para censurar el uso de determinados t6rmmos, por ejemplo en un foro o incluso en la aplicación Smart Form. De nuevo, puede utilizar funciones de cadena o funciones de expresión-lar con este fin. --a0-

La función de cadena que más se suele utilizar para realizar sustituciones es str_replace (1. A continuación se recoge su sintaxis:

```
aixed atr_replace(aixedaguja,  aixedaueva_aguja, aixedJHjar(, int i.ndmerol11;
```

Esta función sustitu~á todas lu instancias de *aguja* en *pajar* por *numi..11guja* y devolverá la nueva versión de *JlllJ"r.* El cuarto parámetro opdonal, naimm, contiene el número de sustitudones realizadas. Se aiiadió en PHPS.

Nota

Desde PHP 4.0.5 puede pasar todos los parámetros como matrices y la función se comportaré de manera bastante inteligente. Puede pasar una matriz de palabras para su sustitución, una matriz de palabras con la que sustituirlas (respectivamente) y una matriz de cadenas a las que aplicar estas reglas. La función devolveré una matriz de cadenas revisadas.

En el formulario Smart Form podrían aparecer palabras malaonuttee incluidas por los ~~os al manifestar sus quejas. Como programadores, póttemos evitar que los ~:bstil\t~ depart.ament~ de Bob reciban los insultos si utilizamos una matriz $off color que contenga una serie de palabras ofemivü:veáñios un e,empio de cómo utilizar str_replace () con una matriz:

```
$feedback•  •tr_replace(Soffcolor,  •11•••,  $feedback);
```

La función substr_replace (I se utiliza para buscar y reemplazar una deter-——da subcadena de una cadena en función de su posición. Su sintaxis es la siguiente:

```
strin9 aubatr_raplace(atrin9 cadea•,  •tring auatitucidn,
        '  inc inicio,  int  (longitud)  |;
```

Esta función sustituirá parte de la cadena *aulm11* con la cadena *sustit~ci6n.* La parte que se.s~tuirá depende de los valores de los parámetros *inicio* y *longitud.*

El valor *1n1~* reJ>~ta el desplazamiento dentro de la cadena en el que comenzará la sustitución. 5l el valor fuera 0 o positivo, el desplazamiento se establecería con respecto al principio de la cadena; si fuera negativo, el desplazamiento se establecería con respecto al final de la cadena. Por ejemplo, esta línea de código "Ulltituiráel último carácter de $test con "X":

```
Steat • eub9tr_repl~ce(Steet, •X•,  ·11;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

El valor *longihul* es opcional y representa el punto en el que PHP dejará de realizar sustituciones. Si no se indica este parúnetro, la cadena será sustituida desde el *inicio* hasta el final de la cadena.

Si el valor del parámetro *longimd* es cero, la cadena de sustitución se insertará dentro de la cadena sin sobrescribir la cadena existente. Una *longitud* positiva representa el número de caracteres que se sustituirán con la nueva cadena. Una *longitud* negativa representa el punto en el que le gustaría detener la sustitu~ de caracteres, contados desde el final de la cadena.

Introducción a las expresiones regulares

PHP admite dos estilos de sintaxis de expresión regular: POSIX y Peri. El estilo POSIX de expresión regular se compila en PHP de manera predeterminada,pero se puede utilizar el estilo Peri compilando en biblioteca PCRE (Expresión regular y compatible con Peri). En nuestro caso, utilizaremos el estilo POSIX que es más sencillo, pero si es programador de Peri o desea saber más sobre la biblioteca PCRE, consulte el manual en Unea dirigifndose·a bt tp : / /pbp. net.

¡;or ló tánto,-iais.éo1néicieñc1u clé patróñ ria1izadas hasta iihorii üilifzabari Eun-- ciones de cadena. Nos hemos limitado a las coincidencias exactas o a la coincidencia exacta de subcadenas. Si desea establecer coincidencias más complejas, debería utilizar expresiones regulares. El uso de las expresiones regulares resulta difícil de comprender al principio pero pueden llegar a ser extremadamente útiles.

Los fundamentos

Una expresión regular es una forma de describir un patrón en un texto. La coincidencia exacta (o literal) que hemos realizado hasta el momento es una forma de expresión regular. Por ejemplo, en una sección anterior buscamos las expresiones regulares "ebop" y "delivery".

En PHP, el establecimiento de coincidencias con expresiones regulans se parece mú a utilizar la función atratr () que a realizar una comparación de igualdad, porque se hace coincidir una cadena con alguna parte de otra cadena (puede ser cualquier parte de la cadena a menos que se especifique otra cosa). Por ejemplo, la cadena "abop" coincide con la expresión regular "sbop". Pero tambifn con las expresiones regulares "b", "bo" etc.

4! Podemos utilizar caracteres especiales pan indicar un metasignificado ademú hacer coincidir caracteres de forma exacta. Por ejemplo, mediante el uso de caracteres especiales se puede indicar que ocurra un patrón al inicio o al final de una cadena, que parte de un patrón se pueda repetir o que los caracteres de un patrón sean de un tipo dado. Tambifn podemos buscar coincidencias literales de caracteres especiales. Examinaremos todas estas posibilidades.

~onjuntos y clases de caracteres

El uso de conjuntos de caracteres aumenta el potencial de las expresiones regulares con respecto a las expresiones de coincidencia exacta. Los conjuntos de earaeteres se pueden utilizar para hacer coincidir cualquier carácter de un tipo dado (en realidad son un tipo de comodín).

En primer lugar, puede utilizar el carácter como comodín pan sustituir cualquier otro carácter individual a excepción del carácter de nueva línea (\n). Por ejemplo, la expresión regular

```
.at
```

coincide con las c_ldenas 'cat •, ·aat · y ·mat •, entre otras. Este tipo de comodín de coincidencia se suele utilizar para búsquedas de nombres de archivo en sistemas operativos.

Sin embargo, las expresiones regulares permiten especificar el tipo de carKter que se desea buscar y, de hecho, se puede especificar un conjunto al que pertenezca un carácter. En el ejemplo anterior, las e,q,resiones regulares coinciden con ·cat · y 'mat ' pero tambim con ·#at ·. Si desea limitar el rango de coincidencias a un carácter entre la a ,¡ l.-z; puede liacerlode la siguiente 1ori:ni:"-·

```
(a-aJ
```

Todo aquello que quede encerrado entre los corchetes ([J) se considerará una clase de carácter (un conjunto de caracteres a los que debe pertenecer un carácter coincidente). Tenga en cuenta que la expresión incluida entre los corchetes coincide con~ único carácter. Puede incluir un conjunto de caracteres; por ejemplo:

```
ruiou~
```

significa cualquier vocal.

Tambim puede describir un rango, como hicimos anteriormente utilizando un guión o un conjunto de rangos.

```
[a-&A-ZJ
```

Este conjunto de rangos equivale a cualquier carácter aUaWtíco ya sea maydsculas o minúsc:ulas. Tambifn puede utilizar conjuntos para excluir un carácter del conjunto. Por ejemplo:

```
1ª•·•1
```

EL_HOLISTICO@HOTMAIL.COM - 2006

equivale a cualquier car6cter que no esté entre la a y la z. El símbolo de acento circunflejo significa *no* cuando se coloca dentro de corchetes. Si se utiliza fuera de los corchetes tiene otro significado, que examinuemos en breve.

Ademú de enumerar conjuntos y rangos, se pueden utilizar varias clases de car6cter predefinido en una expresión regular, que se recogen en la tabla 4.3.

Tabla ,U.ci.... de caracteres para su uso en expresiones regulara de estilo de POS!)(.

Clase	Coincidencia
C C :alnum: I I	Caracteres alfanum6ricoa
C C :alpba: I I	Caracteres aHabétlcos
[[:lower:))	LelrIls en minúsculas
e ['upper: I I	Letras en mayúsculas
e 1,d191t,11	DIgltos decimales
[[:xdigit:ll	DIgitoa hexadecimales
e [:punct:))	Puntuación
[[:blank:)J	Tabuladores y upacloa
[[:epace:))	Esp~cioa en blanco
C I: cntrl: I J	Caracteres de control
e [:print, J J	Todos 1os caracteres lmprimibles
n :graph:°) J	Todos 1oa caractere a ImprimIbles a excepción de espaclo

Repetición

Con frecuencia necesitamos especificar que puede haber varias instancias de úna cadena o clase de car6cter concreta. Para ello, podemos utilizar dos caracteres especiales dentro de una expresión regular. El símbolo • significa que el patrón se puede repetir cero o mú veces y el símbolo + indica que el patrón se puede repetir una o varias veces. El símbolo puede aparecer directamente tras la parte de la expresión a la que se aplique. Por ejemplo:

II:alnua:11•

significa ·al menos un cancter alfanumérico".

Subexpresiones

Suele resultar lltil poder dividir una expresión en subexpresiones para representar, por ejemplo, •al menos una de estas cadenas seguidas exactamente por una de

élas otras". Para ello, se pueden utilizar los paréntesis, exactamente de la misma forma que haria en una expresión aritmética. Por ejemplo:

(very)•large

equivale a ' large ', ' very large', ' very very large •, etc.

Recontar subexpresiones

Podemos especificar cúntas veces se repite un elemento utilizando ~iones numéricas entre llaves ({ }). Podemos mostrar el número exacto de repeticiones ({3} significa exactamente 3 repeticiones), un rango de repeticiones ({2, "} significa de 2 a 4 repeticiones) o un rango sin c:errar de repeticiones ({2, } significa al menos dos repeticiones). Por ejemplo,

(very I (1, 3J

equivale a 'very ', ' very ve~ ' y •very very very '

Anclajes al principio o al final de una cadena

El patrón Ca-zJ comparan todas lu cadena que contengan un cancter alfabético en min11scula. No importa si la cadena sólo tiene un canc:ter o sólo l:IOinc:ide un car6cter dentro de una cadena de mayor tamaño. Podemos especilicar si una subexpresión dada deberla aparecer al final, al principio o en ambos lugares. Esta opción resulta bastante lltil cuando queremos estar seguros de que ea la cadena. sólo aparece el término de búsqueda y nada más. Se utiliza el acento cin:unflejo ("') al principio de una expresión regular para indicar que debe aparecer al prindpio de una cadena buscada. El símbolo $ se utiliza al final de una expresión regular para indicar que debe aparecer al final. Por ejemplo, la siguiente expresión busca coincidencias con bob al principio de una cadena:

La siguiente expresión busca coincidenciascon c011 al final de una cadena:

c-s

Por último, la siguiente expresión busca coincidencias con cualquier canlc:ter único de la a la z, en su propia cadena:

• (a-z)S

Bifurcación

Puede representaruna opción en una expresión regular utilizado una barra vertical. Por ejemplo, si deseamos buscar coincidencias con COII, edu o net, podemos utilizar la siguiente exfresión:

lc-1 I ledu) I laetl

EL_HOLISTICO@HOTMAIL.COM - 2006

Buscar coincidencias de caracteres especiales

Si desea buscar coincidencias con los caracteres especiales mencionados como •, { o $, debe anteponerles una barra invertida (\). Si desea representar una barra invertida, debe sustituirla por dos barras invertidas, \ \.

No olvide incluir patrones de expresión regular en cadenas con comillas simples. El uso de expresiones regulares en cadenas de PHP con comillas dobles aftade complicaciones innecesarias. PHP también utiliza la barra invertida para escapar caracteres especiales, como por ejemplo la propia barra invertida. Si desea buscar una barra invertida en su patnSn, tendrá que utilizar dos ~ra indicar que se trata de un literal, no de un código de escape.

Del mismo modo, si desea una barra invertida literal en una cadena de PHP entre comillas dobles, también tendrá que utilizar dos. El resultado final de estas reglas es que una cadena de PHP que representa una expresión regular que contenga una barra invertida literal necesitará cuatro barras invertidas. El intérprete de PHP analizar las cuatro barras como si se tratara de dos. Tras ello, el intérprete de expresiones regulares analizará estas dos barras como si fueran una.

El signo del dólar también es un carácter especial en las cadenas de PHP entre comillas dobles y en expresiones regulares. Para obtener un $ literal en un patrón, necesitará •\\\$•. Como esta cadena se encuentra entre comillas dobles, PHP la analizará como \$, lo que el intérprete de expresiones regulares puede compara con un signo de dólar.

Resumen de los caracteres especiales

-irnlis tablas4.1-y4:5 sereeoge'urrresumen de todos los taracterenspeciales La tabla 4.4 muestra el significado de los caracteres especiales fuera de corchetes y la tabla 4.5 muestra su significado cuando se utilizan dentro ellos.

Tabla 4.4. Resumen de caracteres especiales utilizados en expresiones regulares POSIX fuera de corchetes.

Carácter	Significado
\	Carácter de escape
	Coincidencia al principio de la cadena
$	Coincidencia al final de la cadena
	Coincidencia de cualquier carácter a excepción del carácter de nueva llnea (\n)
	Inicio de una opción alternativa (como OR)
	Inicio de un subpatrón
	Final de un subpatrón

Carácter	Significado
•	Repetir 0 o más veces
+	Repetir 1 o más veces
	Inicio del cuantificador mln/máx
	Inicio del cuantificador mlnlmáx

Tabla 4.5. Resumen de caractares especiales utilizados en expresiones nigu1ar9s POSIX dentro de con:hetes.

Carácter	Significado
\	Carácter de escape
	NO, solamente si se utiliza en una posición inicial
	Usado pera especiftcar rangos de cará~

Utilizar estos elementos en Smart Form

En la aplicación Smart Form, las expresiones regulares se pueden utilizar con dos fines. El primero consiste en detectar términos concretos en los comentarios envia dos po~ ~os clientes. ~odemos utilizar expresiones regulares para ahorramos ~abaJO.Sl utilizamos funaones-de-cadena, tendremos querelilizar tres-búsquedas sl queremos encontrar 'shop ', • customer service • o • retail • cuando ba~taría con una expresión regular para realizar la misma operación:

```
•bop(cu•ca.er •ervice(recail
```

El segundo uso consiste en validar las direcciones de correo electrónico del" cliente en nuestra aplicación codificando el formato estandarizado de una dirección de correo electrónico en una expresión regular. El formato incluye varios caracteres alfanuméricos o de puntuación, seguidos por el símbolo e, por una cadena alfanumérica y guiones, seguidos por un punto, por más caracteres alfanuméricos y guiones, y posiblemente por más puntos, hasta llegar al final de la cadena, que se codifica de la siguiente forma:

```
•ca-zA-zo-,_\-\.J+ela-zA-Z0-9\-J+\.Ca-zA-zo-,,-,.J+$
```

La subexpresión .. [a-zA-Z0-9 \-\. I+ significa "la cadena debe comenzar. con una letra, un número, un guiónbajo, un guión medio o un punto O una combi· nación de estos caracteres•. Cuando se utiliza un punto al principio o al final de una clase de carácter, pi~rde su sentido especial de comodín y Sé convierte en un punto literal.

EL_HOLISTICO@HOTMAIL.COM - 2006

El sfmboloeequivale al literal•· La subexpresión [a-zA-Z0-9\- J + equivale a la primera parte del nombre del host incluyendo caracteres alfanum4ricos y guiones. Como puede observar se ha quitado la barra invertida delante del guión dado que se trata de un carácter especial cuando va entre corchetes.

La combinación \ . equivale al literal •. Como utilizamos el puntero fuera de Ju clases de carácter, es necesario escaparlo para que sólo coincida con el punto literal.

La subexpresión [a- ZA- Z0- 9\ - \ • J +$ equivale al resto del nombre del dominio, incluyendoletras, números, guiones y más puntos si fueran necesarios,hasta el final de la cadena. Si analiza esta expresión, descubrirá que es posible crear direc· dones no válidas equivalentes a la expresión regular. Resulta prácticamente imposible capturar todos los casos, pero hemos logrado mejorar bastante la situación. Puede volver a definir esta expresión de muchas formas. Por ejemplo, puede enu· merar todos los TDL válidos. Ahora bien, es necesario tener cuidado a la hora de restringir elementos ya que una fundón de validación que rechace un l'l' de los datos válidos resulta mucho más molesta que una que permita un 10,C, de datos no válidos. Una vez analizadas las expresiones regulares, ya podemos centramos en las funciones de PHP que las utilizan.

Buscar subcadenas con expresiones regulares

Í.a búsqueda de subcadenas es la aplicación principal de las exp~esiones regula· res que acabamos de desarrollar. Las dos funciones disponibles en PHP para buscar coincidencias con expresiones regulares de tipo POSIX son ereg (1 y eregi (1. La sintaxis de la función eregO es la siguiente:

```
int ere9 (scring ¡i.crda, string bdaqueda, array tco.t.acide.aciaal) 1
```

Esta función busca la cadena *bústfuala* para encontrar coincidencias con la expre· sión regular *J18tr6n.* Si se encuentran coincidencias para las subexpresiones de .,,.. *tr6n,* se almacenarán en la matriz *coincidatcias,* un subexpresión por cada elemento de matriz. La fundón eregi ( ) es idfntica con la salvedad de que no discrimina entre mayúsculas y minúsculas. Podemos adaptar el ejemplo Smart Form para utilizar expresiones regulares de la siguiente forma:

```
if  Ctere,¡t(••ta·sA·Z0·9_\·\.J+eCa·&A·Z0·9\·l+\.[a·&A·Z0·9\·\.J+S',      S... 1111
{
   ecbo •nac ta aot a vaUd  ...11 addr•••·   Pl•••• retazD to tlla'
         .' previoua paga a.ad try again. ';


Stoeddreaa • 'feedbac~je11plo.c011''1     // el valor predeterminado
U  (ere,il'•boplcaatGMr' Hrvicelretail',  $feedback1 1
   Sto.ddz9•• •  •ntail ..jemplo.c0II' '1
•1•• if (er..,il'deliver.•lfulfil.••,     $feedback))
   $ta.ddraN  •  'fulfil___t ..jemplO.C011''1
•1•• if (er--,il'billjacco1111t•,  $feedback))
   $toaddreu  •  'account• ..jemplo.coa•'1
```



```
if leregi1bigcuato-.r\.c011',    S... 1111
     $toaddre•• •  'bobeejemplo.c011 · 1
```

Sustituir subcadenas con expresiones regulares

Tambifn podemos utilizar expresiones regulares para buscar y reemplazar subcadenas de la misma forma que utilizamos •tr _replace ℓ). Las dos fundones disponibles para realizar esta operaci6nsonereg_replacO y eregi_replace (). La función ereg_replace () tiene la siguiente sintaxis:

```
•t~iag ereg_replace(string ¡i.trda, string •uat.ttuc.td.a.  •tring bdaquedal;
```

Esta función busca la expresión regular JMtrdn en la cadena *búsquaill* y la reem· plaza con la cadena *sustitución.* La función eregi_replace () es idintica con la diferencia de que no discrimina entre mayúsculas y minúsculas.

Dividir cadenas con expresiones regul;¡res

Otra función de expresión regular útil es apli t(l, cuya sintaxis e la siguiente:

```
array split(striag  ¡i.tr6a, •tring bdaqueda, lat  [ ~JI;
```

Esta función divide la cadena l,ústflUIU en subcadenas en la expón regular *p11tnm* y devuelve las subcadenas en una matriz. ..Ji

El valor entero mlfz limita el número 4e eleme~tos ql,le p1,1~e Jncluir la matriz. Esta fundón resulta útil para dividir nombres de dominio o fechas. Por ejemplo,

```
$addn•• •  •u• ~ J nplo.c0II' 1
$arr • •plit  |'\.je,    $addres•1l
wbile 1list($key,   $Yalue1 • each 1$arr1l
   ecbo •cbr />'.$Yalue1
```

Esta función divide el nombre del host en cinco componentes e imprime cada uno en una línea distinta.

Comparar funciones de cadenas y funciones de expresiones regulares

En general, la ejecución de funciones de expresiones regulares resuita menos eficaz que las fundones de cadena con fundonalidad similar. Si su aplicación es sencilla, utilice expresiones de cadena.

EL_HOLISTICO@HOTMAIL.COM - 2006

Lecturas adidonales

PIIP-- •-..;Inrf1'M. ' 211:id• *tmcimm* deadlm lll'* CllnllalD hl.-
d6n•e11,aclliWalcll'tlcD).cm_.el-..l•-dePHPpuaa,..-,li
LaCI , WTll¡l ....
pw ... 1&-,.. •la nflUP 11 9lli ,'''' •• UNIX.
T..... S.cali ~111er~c:c-.
a..a-1111ow••2'.wl.•....,._fwidllde,..Vf dGe,-acanw
•s1ht, ... I ¡ll) 'f..... la~m ..... Sit ....
..... Ilail.Yal() , ..., • Jthe~q,://ww.:aead.cc./codex.pbp'Ud.
1a-1ngi.-L
Lacmncta.....,..... ... ..,. tiempo.C.....
..... e ,.,.,....._ ....waHU f., ...

A con.tinuad6n

_. . .iil r e le; ..... , ...ut111ar,..,.....
..._,.. 1 ....._., ...de.-.. :jo'lp¡.¡ 1 ffllnr la _ 1 le lll atillm c6dlp
1•r iH t

5

Reutilizar código y crear funciones

En este capitulo explicaremos cómo la reutilización del código permitedesarrollar código más coherente, fiable y sencillo de mantener con menos esfuerzo. Mostraremos técnicas para dividir el código en módulos y reutilizarlo. ~ ello, comenzaremos por explicar el sencillo uso de las instrucciones r~quire (l e include (l -que permiten utilizar 'el misíno código en varias páginas. Explicaremos por qué resultan preferibles a las inclusiones en el lado del servidor. En el ejemplo que presentaremos se recurre a archivos de inclusión para lograr un aspeeto visual y operativo uniforme en todo el sitio. Explicaremos cómo escribir funciones propias y llamarlas utilizando funciones de generación de páginas y formularios como ejemplos. En este capítulo abordaremos los siguientes aspectos:

- Reutilizar código
- Utilizar require () e include ()
- Introducción a las funciones
- Definir funciones
- Utilizar parámetros
- Ámbito
- Devolver valores
- Uamadas por refere:lcia frente a llamadas po~ valor
- Recursión

EL_HOLISTICO@HOTMAIL.COM - 2006

Reutilizar código

Uno de los objetivos de los ingenieros infonúticos consiste en reutilizar código para no tener que escribirlo de nuevo; No porque se trate de un grupo especialmente vago sino porque la reutilización de código contribuye a reducir los costes, a aumentar la fiabilidad del código y a mejorar la uniformidad de los resultados. Lo ideal seria que los nuevos proyectos se crearan combinando componentes de código ya existentes, con un mínimo de desarrollo desde cero.

Costes

Durante la vida útil de un fragmento de código, se invierte más tiempo en modificarlo, probarlo y documentarlo que en crearlo. Si est4 escribiendo código comercial, debería intentar limitar el número de lineas que se utilizan dentro de la organización.

Una forma pr,ctica de lograrlo consiste en volver a utilizar código ya creado en lugar de escribir versiones ligeramente diferentes del mismo código para realizar una nueva tarea. Menos código significa menos costes. Si ya existe software que satisface sus necesidades para un proyecto, adquiéralo. El coste de comprar software existente es siempre inferior al coste de desarrollar un pn>4ucto equivalente. Ponga mucha atención si el software existente no cubre exactamente sus necesidades :n.que su modificadón podrfa resultarmás dificil que su creación desde cero.

Fiat>flidad

Si un módulo de código se utiliza en otra parte de la organización, es muy probable que se haya probado exhaustivamente. Aunque se trata de unas pocas lineas, si se vuelve a escribir, es probable que pasemos algún elemento por alto incorporado por- su autor original o que se agregó al código tras descubrir un fallo durante la fase de prueba. El código existente suele ser más fiable que el código recién creado.

UJ1iformidad

Las interfaces externas de nuestro sistema, incluidas las interfaces de usuario y las interfaces para los sistemas externos, deben ser uniformes. Para escribir código nuevo que resulte uniforme con respecto a la forma en la que funcionan el resto de las partes del sistema se necesita una voluntad y un esfuerzo deliberado. Si reutilizamos código ejecutado en otra parte del sistema, la uniformidad quedan garantizada autolftjticamente.Sobre todas estas ventajas, la reutilización del código significa menos trabajo, siempre y cuando dicho c6digo sea modular y esté bien

escrito. Al trabajar, intente reconocer secciones de su código a las que pueda llamar en el futuro.

Utilizar require() e include()

PHP incorpora dos instrucciones muy sencillas pero de gran utilidad para permitir la reutilización de cualquier tipo de código. Mediante el uso de las instrucciones requiere() o include () puedecargarunan:hivoensusecuenciadecomandos de PHP. El archivo puede contener todos aquellos elementos que se incluirían por regla general en una secuencia de comandos, incluidas instrucciones de PHP etiquetas HTML, funciones de PHP o clases de PHP. Estas instrucciones funciona'n de manera similar a las inclusiones del lado del servidor que ohecen muchos servidores Web y a las instrucciones #include de C o C++.

Utilizar require()

El siguiente códillo se almacena en un archivo llamado reusable . php:

```
c?pllp
   ecbo 'B•r• i• a -ry   •imple PBP 8l:.i —— DIC..cbr />'1
?>
```

!l siguiente código iie almacena en un archivo lwnado iliilfii. pllp:

```
c?pbp
   eclio '''•Tliia l.   ll.b•'•iia flie~,-bi,7>.-,   ------- ------ ....
   requirel •re1U1able.pbp' );
   ecbo 'Tbe •cripl. vill elld now.cbz; />';
?>
```

Si carga reuHble. pbp, es probable que no le sorprenda ver aparecer la secu~ "Here i• a very simple PHP statement. • en el navegador. Si carga el an:hivo main. php ocumrá algo un tanto mú interesante. El resultado de esta secuencia de comandos se ilustra en la figura 5.1.

Se n~t~ un archivo ~ara utilizar la instrucción require O . En el ejemplo anterior, utilizamos el archivo reusable. pbp. Al ejecutar nuestra secuencia de comandos, la instrucción requi re ()

```
require( 'reu•able.pbp•  l;
```

se sustituye por los contenidos del archivo solicitado y, a continuación, se ejecuta la secuencia de comandos. Esto implica que al cargar main. php, se —ta como si la secuencia de comandos estuviera escrita de la siguiente forma:

```
c?pllp
   echo 'Tbi• l• ll.ta.' .. in file.cbr  />';
   •ello '••r• i•  a Tery  • ~ le ... •tat...at.cJtr  />',
```

```
   ecbo 'Tlle •cripc will ead now.cbr />';
?>
```

Al utilizar require (| es necesario tener en cuenta las diferentes formas en las que se procesan las extensiones de nombre de archivo y las etiquetas de PHP.

Figura 5.1. El resultado de maln.php muestra el resultado de la Instruc:c:ión require().

Extensiones de nombre de archivo y requíreü

PHP no examina la extensión del nombre de archivo en el archivo requerido. Esto significa que puede asignar el nombre que desee a su archivo siempre y cuan-<jo no tenga previsto llamarlo directamente. Al utilizar require (| para cargar el archivo, se convertirá de manera efe!=tivaen parte de un archivo de PHP y se ejecutará como tal.

Por regla general, las instrucciones de PHP no se procesarán si estuvieran en un archivo llamado, por ejemplo, page. html. Sólo se llama a PHP para analizar archi-VOS ardefinitexterisiónes como :php (aunque en su áréhivo de c:ónfiguración de servidor Web puede ser diferente). Sin embargo, si carga la página page. html a través de una instrucción require (,), se procesarán todas las instrucciones de PHP incluidas en su interior. Por lo tanto, puede utilizar cualquier extensión que desee para archivos de inclusión, pero conviene utilizar una convención lógica, como . inc o . php.

Recuerde que si almacena los archivos que terminan en . inc o alguna otra extensión no estándar en el árbol de documentos Web y los usuarios los cargan directamente en el navegador, podrán ver el código en forma de texto sin procesar, incluidas las contraseñas. Por lo tanto, es importante almacenar los archivos de inclusión fuera del árbol de documentos o utilizar extensiones estándar.

Etiquetas de PHP y require()

En nuestro ejemplo, el archivo reutilizable (reusable. php) se esaibe de la siguiente forma:

```
c?pllp
   ecbo 'Her• i• a very •imple PRP •cace..nc.cbr />';
?>.
```


El código de PHP se coloca dentro del archivo en etiquetas de PHP. Necmitad-realizar esta operación si desea que el código PHP incluido en el an:hivo requerido sea tratado como código de PHP. Si no abre una etiqueta de PHP, su mcligo tem tratado como texto o código HTML y no se ejecutará.

Utilizar require() para plantillas de sitios Web

Si las páginas Web de su compañfa presentan un aspecto visual y operativo uniforme, puede utilizar PHP para agregar la plantilla y los elementos estándar utilizando require (| .

Por ejemplo, las páginas del sitio Web de la compañía TLA Consulting presentan la apariencia que se muestra en la figura 5.2. Cuando se necesita crear una nueva página, el desanollador puede abrir una existente, recortar el texto del archivo e introducir el nuevo texto, y guardar el archivo con otro nombre.

Figura 5.2. El sitio Web de TLA Consulting presenta un especto visual Y operativo esléndar en todas sus páginas.

Considere la siguiente situación: imagine un sitio Web que lleva mucho tiempo activo y consta de decenas, centenas o miles de páginas con un estilo común. Ent°"?-ces se toma la decisión de cambiar parte de su aspecto. Puede tratarse de un cambio pequefto, como agregar una dirección de correo electrónico al pie de página de todas las páginas o incluir una nueva entrada en el menú de navegación. Seguro que no le apetece~plicar este pequeño cambio a todas las páginas.

La posibilidad de reutilizar secciones de código HTML comunes a todas las páginas es una opción mucho más práctica que cortar y pegar los cambios en decenas, centenas o miles páginas.

EL_HOLISTICO@HOTMAIL.COM - 2006

En el listado 5.1 se recoge el c6cligo fuente de la p6gina de inido (hCIN.html) ilustrada en la figura 5.2.

U.tado 5.1. llome.lltml. Código HTML de la~ de Inicio de TIA Conlultlng.

```
cbr.111,.
cbau,.
   ccicle,.11.r.  Con•ulcing Pty LCdc/cicle,.
   c•Cyle,.
      b1 {color,wbice;  fonc··ise:2tpc,  cexc·align,c•ncer,
         fODC·f&llily:arial,•an····rif)
      .llellu {color,wbiCe;  fODC··ise,12pC; cezc·ali911:cencer1
         fODC·faaily:arial,•ane····rif;  fonc-weigbC:bold)
      cd {backgrowul:black)
      p {color:black; fanc-eise:12pc;  cexc·align:juecify;
         fODC·faaily,arial,•&ne····rif)
      p.fooc {color:wbice,  fODC··ise,tpc; cezc-align:cencer,
         f0DC·faaily:arial,•&11e····rif1 foDC·weigbc:bold)
      a,liDk,a,wieiCed,a:acciwe {color:wbice)
   c/•Cyle,.
c/bead,.
cbodp

   el··  page 11.eader  ··>
   cl:11111• widCb•"lDDt•  cellpadding••12•  cell•pecing••D•  border•"D">
   ccr bgcolor-•black">
      ccd aliga••left•>cill!I 11rc•"logo.9U• alC••TLa logo•  beigbl:•°'70•
widCbi-₀'70°></Cd>
      --cCd>
         cbl>TLI'. ConeulCingc/bl>
      c/Cd>
-- ctd ¡,,11p;;··rigbt°₀>cim,g·1rc.;•logo:9i!•  alc••TLA 1ogo•-laei9bc••'7o•
widCb•°'7D°>c/Cd>
      c/Cr>
      c/CUle>

   el••  —DU  ••>
   ccable widCb•"lDDI°  bgcolor••wbice•  cellpedding••t•  cellepacing••t•>
   ctr  >
      cCd wideb•"251">
         ciag arc•"••logo.gif•  ale•••  beigbl:•"20"  widcb-•20•,.
         cep&D cla•••°llellU°>Rollec/apan>c/Cd>
      ccd widCb•"251">
         ciag arc•"••logo.gif•  ale•••  beigbl:•"20°  widch•"20">
         capan cla•••"i99Du°>CODCaccc/•PaJa>C/Cd>
      cCd widCb•"251">
         ciag arc•"••logo.gif•  ale•••  beigbc.•20•  widCh•"20">
         capan cla..•°llellu°>lerwice•c/mpa11>c/Cd>
      cCd widCb-₀251°>
         ciag •rc···~·logo.gif•  ale•••  beigbl:•"20°  wideb•"20">
         'ca¡IIID cl..••"llellU°>liCe  llapc/llJl&D>c/Cd>
   c/tr>
   c/tüle>

   el··  ,-ae  concenc  ··>
```

```
   cp>..lcaae co Che baae of TLA COD9Ulcing.
   PlHM  Cake aaae ci-  co gec co ~    u. c/p>
   cp>Ve epecialise iD Mning  your buineH    Meda
   and  bope  to bear fraa yau  —— .c/p>

   el··  page footer  ••>
   ctable widtb-•1001•  bgcolor•"black•  c:ellpadding•"12°  bordllr•"D°>
   ctr>
      ctd>
         cp cla••••foot">6Copyl  TLA  Conaulting Pe.y  Ltd.c/p>
         cp cla•••foot">Plea••  •••  our
            ca bref•"legal.pbp°>legal  iDforaatioD pa11ec/a>c/p>
      c/td>
   c/tr>
   e/table>
c/body>
c/btab
```

En el listado 5.1 puede ver que el archivo consta de wi número de secciones distintas. El encabezado contiene las definiciones CSS (del inglés, Cascading Style Sheet, Hoja de estilo en cascada) utilizadas por esta p6gina. La sección titulad.a "page header" (encabezado de p6gina) muestra el nombre de la ~mpatlía y él logotipo, la sección "menu" (menú) crea la barra de navegac:i6nde la p6gina y la sección "page content" (contenido de p6gina) integra el texto exclusivo de esta p6gina. A continuación, aparece el pie de p6gina. Podemos dividir este archivo y designar las distintas partes como header. inc, home.php y footer. inc. Las seccionesheader. inc y footer. inc contendrin código que se reutilizar6 en Otl'Up4gimls.

El archivo home.php sustituye a hou.html y contiene el texto exclusivo de esta p6gina y dos instruccionesrequire (). como se muestra en el listado 5.2..

Ustado 5.2. llome.pllp. C6cllgo PHP que producela p6glna de Inicio de TLA.

```
c?pbp
   require ('beader. inc'I ;
?>
   el··,...   CODCeDt  ··>
   cp>Vel-  to tbe ~    of TLA Couulting.
   Pleaae take aaae ti-  to get to kllow u.c/p>
   cp>N• apecialise iD aening  your buain••• 11eed•
   ud  llape  to hear fr-   yau  ._.c/p>
c?pbp
   require('footer.inc'I;
```

La instruc:cic5n. require O de home.php carga los archivos header. inc y footer. inc.

Como se mendonó, el nombre asignado a estos archivos no afecta a la forma en la que se procesan al llamarloscon require (). UnaconYend6rhabituaJ.awique del todo opdonal, consiste en llamar a archivos puciales que terminum incluidos en otros archivos algo. inc (aquf inc equivale a incluir). Tambim resul~ habi-

EL_HOLISTICO@HOTMAIL.COM - 2006

tual, adem6s de conveniente, colocar los archivos que se incluirm dentro de un directorio al que puedan ac:cedersus secuencias de comandos pero no permitir que se carguen individualmente a través del servidor Web. De esta forma impediremos su carga, lo que dar6 lugar a: a) que se produzcan errores si la extensión del archivo es .php pero sólo contienen una p6gina o secuencia de comandos parcial o b) permitir que se pueda leer el código fuente si ha utilizado otra extensión.

El archivo header. inc contiene las definidones de CSS utilizado por la página, las tablas que muestran el nombre de la compal\fa y los menús de navegación como se ilustran en el listado 5.3.

us .. do 5.3. huder.Inc. El encabezado reutilizable para todas las"péginas Web de TLA.

```
cbt•l>
cbead>
   etitle>TLA c:on.ulting Pty Ltdc/title>
   catyle type•"te&t/c••">
      bl (color,wbit•1 font·•ise,24pt1 text-align,center,
         font·faaily:arial,aan.·••rifJ
      ....u (color:wbit•1 faat·eise,12pt1 text-align:center;
         faat-faaily,arial,11a11•· ..rif; font-weigbt1boldJ
      td (background,blackJ
      p (color:blaclc; font·•ise,12pt, tezt-align,juatify;
         font·fallily:arial,•aa.- ..rifJ
      p.foot (color,wbite; foat-•ise,tpt1 tezt-align,center,
         font·faaily,arial,aan.·••rif; font·weigbt:boldJ
      a:link,a,vi•ited,a,active (color,wbiteJ
   c/•tyle>
c/bead>
cbody>

   el•• page baadar ··>
   ctable widtll-"1001" callpadding••12• cellapacing··0• border•"0">
   ctr bgcolor•"black">
      etd align••left">ci119 •rc•"logo.gif• alt••TLA logo• beight•"70"
width•"70">e/td>
      ctcl>
         cb1>TLA Coa.ultingc/bl>
      c/td>
      etd align••rigbt">ci119 arc••logo.gif• alt••TLA logo• baigbt•"70•
vidth•"70°>c/td>
   c/tr>
   e/table>

   el·· —u··>
   ctable width•"1OOI" bgcolor•"wbite• cellpadding••t• cell•pacing•"t">
   ctr >
      ctd widtb•"251">
         ci119 •rc•"•·logo.gif• alt•"" beight•"20" width•"20">
      • cmpan clna·•—u•>—c/apan>c/td>
      ctd widtll-o251">
         cltag •rc•••·logo.gif" alt••• baigbt•"20" wideh•"20">
         c.... cl••••"mellll">Ccxltacte/....>c/td>
      ctd widtll•"251">
```



```
         cimg •rc•"••logo.gif• alt•"" beigbt•"20" widtb•"20"•
         c•pan cla••••..nu">Serviceac/!lp&D>c/td>
      ctd widtb•"251">
         ei11g •rc•"••logo.gif• alt•"" baight•"20" width•"20">
      e•paa cla•••"menu">Site llapc/..-n>c/td>
   c/tr>
   e/table>
```

El archivo footer. inc contiene la tabla que muestra el pie de página en la parte inferior de cada página. Este archivo se muestra en el listado 5.4.

UstNo 5.4. rooter.Inc. El ple de péglna '9UIIIIzable para todas las ~ Web de TLA.

```
el·· page footer -·>
   <table widtb•"lOOI• bgcolor•"black• cellpadding•"12" border•"0">
   ctr>
      etd>
         cp cla•••"foot">6copy, TLA C0119ultins, Pty Ltd.e/p>
         cp cla••··foot">Pl..•• ••• our ca hr•f•"legal.php">legal
          infoEIDlltion pagec/a>c/p>
      e/td>
   c/tr>
   e/table>
c/body>
e/ht•b
```

Este enfoque facilita la tarea de lograr una apariencia uniforme en 1 In sitio Web y permite crear una página Web con el mismo etilo escribiendo algo qkamo:

```
c?pbp requirel'baader.inc:'l1 ?>
llera i• tbe content for tbia paga
c?pbp raquirel'footer.inc')1 ?>
```

Pero incluso tras crear una gran cantidad de páginu utilizando este encabeudo y este pie de página, resulta sencillo modifiCJr sus respectivos archivos. Con independencia del tipo de cambio que desee realizar (una pequeña variad6n o modificar el diseño completo del sitio), sólo necesitar6 realizar el cambio una vez. N9 tendr6 que variar cada página del sitio porque todu ellu se cargan en los archivos de encabezado y pie de página.

El ejemplo visto utiliza únicamente HTML en el cuerpo, en el encabeZldo y en el pie de página. Pero no tiene por qué ser así. Dentro de estos archivos, podríamos utilizar instrucciones de PHP para generar dinámicamente pertes de lu páginu.

Si desea que un archivo se procese como archivo de texto sencillo ó como H'JML, y no se ejecute código de PHP, puede utilizar la fundón readfile (l, que reproduce el contesiido de un archivo sin analizarlo. Puede ser una importante medida de seguridad si utiliza textos proporcionados por el usuario.

Utilizar include()

Las instrucciones require (l e include<l son pr6cticamente idmticu. La 6nica diferencia entre ambu es que cuando fallan, la instrucción require(l de-

EL_HOLISTICO@HOTMAIL.COM - 2006

vuelve un error terminal, mientras que la instrucción include () devuelve una advertencia.

Utilizar require once() e include once()

Existen dos variaciones de require () e include (): require once () e include_once (). La función de estas construcciones es garantizar que el archivo incluido sólo se incluye una vez. En los ejemplos que hemos desarrollado hasta el momento, encabezados y pies de página, no resulta especialmente útil.

Sí lo es cuando se utiliza require () e include () para incluir biblioteca de funciones. Mediante estu construcciones evitamos incluir inadvertidamente la misma biblioteca de funciones dos veces, lo que volvería a definir lu funciones y generada un error.

Utilizar auto prepend file y auto_append_file

Si queremos utilizar la instrucción require () para agregar el encabezado y el pie de página de todas las páginas, existe otra forma de hacerlo. Dos de Ju opciones de configuración del archivo php. incson auto_prepend_file y auto_append_file. Si asignamos estu opciones a los archivos de encabezado y pie de página, estaremos seguros de que se cargarán antes y despuá de cada página. Los archivos incluidos mediante estu directivas se comportan como si se hubieran añadido por medio de una instrucción include<>; es decir, si falta un archivo, se genera una advertencia. En Windows, la configuración presentará este upecto:

```
---to--_prepend.:.U·l•-- •e, / in•tpub/include/bead•r,inc.__ ---·
auto_append_file• •c,/inetpub/include/footer.inc•
```

En UNIX, la configuración presentará este aspecto:

```
auto_prepend_fil•• "/ho••lu•ern.../include/be•cter.inc•
a,aco_append_file • "/hoae/uaerna../include/footer.inc•
```

Si utiliza estu directivas no necesitará escribir Ju instrucciones require (), pero los encabezados y pies de página dejarán de ser opcionales.

Si está utilizando un servidor Web Apache, puede cambiar varias opciones de configuración como átu en directorios individuales. Para ello, debe configurar el servidor para permitir que el archivo o archivos de configuración principales sean sustituidos. Para configurar las funciones de anexión al principio o al final de un directorio, cree un archivo llamado .htaccess en el directorio. Este archivo debe contener lu dos siguientes líneas:

```
pbp_value auto_prepend_fil• •/~/uae=-/include/lutader.inc•
pbp_value auto_append_fil• •/hoae/uaernaae/includa/footer.inc•
```

Pfjese en que la sintaxis resulta ligeramente diferente con respecto a la misma opción en php.ini as1 como el elemento php_value situado al principio de la

linea. No hay signo igual. También puede alterar otra serie de parámetros de configuración de esta forma.

La posibilidad de establecer opciones en el archivo .htaccess en lugar de hacerlo en el archivo php. ini o en los archivos de configuración de su servidor Web brindan una gran versatilidad. Puede alterar los parámetros en un equipo compartido que sólo afecte a sus directorios. No es necesario reiniciar el servidor Web ni disponer de acceso de administrador. Una de Ju desventajas del mitoclo .htaccess es que los archivos se leen y se analizan cada vez que se solicita un archivo de dicho directorio en lugar de uno al inicio, lo que afecta al rendimiento.

Utilizar las funciones de PHP

La mayoria de los lenguajes de programación constan de funciones. &tas se utilizan para separar el código que realiza una tarea simple y bien definida. Lu funciones facilitan la lectura del código y nos permiten volver a utilizarlo cuando necesitamos realizar la misma tarea. Una función es un módulo de código autónomo que establece una interfaz de llamada, realiza alguna tarea y, i,pcionalmenie, devuelve un resultado. Ya hemos visto una serie de funciones. En los capitulas anteriores, hemos llamado a varia funciones integradas en PHP. También hemos creado algunas funciones sencillas y se han comentado. En esta sección vamoe a analizar en mayor detalle las operaciones de llamar y desarrollar funciones.

Llamar funciones

La siguiente línea representalia"Iliimada más sencilla auna función:

```
II0llbr•_funci6n(I;
```

Esta línea llama a una funciónde nominada nombre_funci6n () que no requiere parámetros. Esta línea de código ignora cualquier valor que pudiera devolvetla función.

Existen varias funciones que se llaman de la misma forma. La funciónphpinf () suele resultar útil para realizar pruebas porque muestra la versión de PHP instaJada, información sobre PHP, la configuración del servidor y valores sobre varias variables de PHP y de servidor. Esta función no toma parámetros y por regla general ignora el valor devuelto. Lu llamadas a phpinfo () presentan este upecto:

```
p!ipinfoO
```

La mayor parte de las funciones requieren uno o más parámetros, que son Ju entradas de Ju funciones. Los parámetros se pasan colocando los datos o el nombre de una variable con los datos incluidos dentro del paréntesis tras su nombre; La llamada a una función con un parámetro presenta este aspecto:

```
DOllbre_funci6n(•parlllatro•1,
```

EL_HOLISTICO@HOTMAIL.COM - 2006

En ese caso, el parámetro utilizado es una cadena que contiene únicamente la palabra parámetro, pero las siguientes llamadas también son correctas según la función:

```
ftOllbre_funci6aC21;
nClllbre_funci6n(1.11J1;
ftOllbre_funci6n(Svarial>lel:
```

En la última linea, $variable puede ser cualquier tipo de variable de PHP, incluso una matriz. Un parámetro puede ser cualquier tipo de datos pero las funciones concretas suelen necesitar tipos de datos dados.

Para saber cuántos parámetros tomó una función, qw! representa cada uno de ellos y el tipo que necesitan, basta con consultar su sintaxis. Al describir las funciones solemos incluir su sintaxis. A continuación, se recoge la sintaxis de la función fopen():

```
reaource fopen( at:riag nombre_de_arcll1vo, 81:riag modo
        (, bool uaar_ruta_lncluaI6n (, reaource zcoateztoJJI
```

Esta sintaxis nos indica varias cosas y es importante saber interpretarlas. En concreto, el término resourc:e que aparece delante del nombre de la función indica que devolverá un puntero (un puntero de archivo). Los parámetros de la función se incluyen dentro de los paréntesis. En el caso de fopen (1 tenemos cuatro. El nombre del archivo y el modo son cadenas, el parámetro *usar_ruta_inclusi6n* es un valor Booleano y *zconttxto* es un puntero. Los c:orc:hetes que rodean a *usar_ruta_fnclusi6n* y:m,ntr.rto indican que se trata de panlmetros opcionales. Podemos incluir valores o ignorar el parámetro. en cuyo caso se utilizará el predeterminado. Sin !!lbargo,.!!ft~nci~nn. .con más ~e u.ll paráJne~C?.~~<H!_al,~Io~ eu~c:te~ignorar los situados a la derecha. Por ejemplo, al utilizar fopen (1, podemos ignorar *zcontexto* o tanto *usar_ruta_inclusi6n* como *zcontexto,* pero no *usar_ruta_inclusi6n* y proporcionar *zcontnto.* Tras examinar la sintaxis de esta función, sabemos que el siguiente fragmento de código es una llamada válida a fopen (1:

```
Sna.. • '11Yfile.tx1:•;
Sopenmode • 'r•,
Sfp • fopaa.(Sna... Sopenmodel
```

Este código llama a la función fopen (1. El valor devuelto se almacenará en la variable $fp. Hemos optado por pasarle una variable llamada $name que contiene una cadena que representa el archivo que queremos abrir y una variable llamada SopenlllOde que contiene una cadena que representa el modo en el que queremos abrir el archivo. En cuanto al tercer y cuarto parámetros opcionales, hemos decidido no utilizarlos.

Llamar a una función no definida

Si intenta llamar a una función que no existe, obtendrá un mensaje de error como el ilustrado en la figura 5.3.

Figura 5.3. Este mensaje de error es el resultado de llamar• una función que no existe.

Los mensajes de error que PHP devuelve suelen ser de gran utilidad. &te nes indica exactamente en qué archivo tuvo lugar el error, en qué linea de la secuencia de comandos y el nombre de la función que intentamos llamar. De esta forma resultará muy sencillo buscar el error y corregirlo.

Si se recibe este mensaje de error, hay que comprobar dos cosas:

- ¿Está escrita correctamente la función?
- ¿Existe la función en la versión de PHP que está utilizandllr

No siempre resulta senc:il\o recordar có1119 se escribe una funtión. Por ejemplo. algunos nombres de función formados por dos palabras utilizan,.,.. guión bajo para separarlas y otras no. La función stripslashes (1 se compm,,.de dos palabl'ti juntas mientras que la función strip_tags(1 util~ un¡u.ió9 jopara separar sus dos palabras. Si se escribe incorrectamente el nombre de ~ función en una llamada se devolverá el error ilustrado en la figura 5.3.

Muchas de las funciones-que-utilizamosen este libro no éxisten en PHP4-ya q,a asumimos que está utilizando la versión PHPS. En cada nueva versión se definen nuevas funciones por lo que las nuevas funcionalidades y el mayor rendimiento justifican la actualización. Si desea saber cuándo se ha agregado una función concreta, puede consular el manual en linea. Si intenta llamar a una función no declarada en la versión que esté ejecutando obtendrá como resultado un error como el que se ilustra en la figura 5.3. Este mensaje de error también puede aparecer si la función invocada forma parte de una extensión de PHP que no se haya cargado. Por ejemplo, si intenta utilizar funciones de la biblioteca GD (manipulación de imágenes) y no la ha instalado, se generará este mensaje de error.

Funciones y el uso de mayúsculas y minúsculas

Recuerde que las llamadas a funciones no discriminan entre mayúsculas y minúsc:ulas por lo que las secuencias func:tion_naae (1, Function_Name() o FONCTION_mum(1 son igualmente v4Udas y devuelven el mismo resultado. Puede utilizar el patrón que le resulte mú sencillo de leer pero debe intentar mantener cierta unüormidad. En este libro, así como en la mayor parte de la documentación de PHP, se utilizan las minúseulas como convención.

EL_HOLISTICO@HOTMAIL.COM - 2006

Es importante recordar que los nombres de funci6n se comportan de manera diferente a los nombres de variable. Los nombres de variable discriminan entre mayúsculas y minúsculas, por lo que $Na.. y $nalle ser6n dos variables diferentes mientras que Name ( ) y name ( ) identifican la misma función.

¡Por qué debería crear funciones propias?

En anteriores capltulos hemos visto ejemplos de cómo utilizar algunas de las funciones incorporadas de PHP. Sin embargo, la verdadera pott!ncia de un lenguaje de programación se demuestra en la capacidad para crear funciones personalizadas. Las funciones incorporadas de PHP nos permiten interactuar con archivos, utilizar una base de datos, crear gnficos y establecer conexiones a otros servidores. Sin embargo, a lo largo de su vida profesional descubrir6 muchas veces que necesita realizar una operación no prevista por los creadores del lenguaje.

Afortunadamente, PHP permite crear funciones para realizar cualquier tarea deseada. Con toda probabilidad, su código combinan funciones existentes con lógica propia desarrollada para realizar una tarea. Si est6 escribiendo un bloque de código que probablemente desee utilizar de nuevo en varios puntos de una secuencia de comandos o en vuias secuencias de comandos, es aconsejable que declare did\o bloque en forma de función.

(a declaración de una función permite utilizar el código creado de la misma forma que las funciones incorporadas. Bastar6 con llamar a la función y definir los parimetros necesarios, lo que implica que puede llamar a la función y voiver a utilizarla vuias veces en la secuencia de comandos.

Estructura básica de una función

La declaración de una función crea o declara una nueva función. La declaración comienza con la palabra clave function, incluye el nombre de la función, los parimetroa obligatorios y contiene el c6digo que se ejecutar6 cada vez que se llame a la función. A continuación, se recoge el ejemplo de una función trivial:

```
functioa my_fuac:tioal)
(
    ecbo 'Ny fuac:tioa wuo call-4•  l
)
```

Esta declaración de función comienza con el término function, para que los lectores humanos y el analizador de PHP sepan que lo que viene a continuación es una~ definida por el usuario. El nombre de la función es my_function. Podemos llamar a nuestra nueva función con la siguiente instrucción:

```
IIY_fucti-CJ /
```


Como habri supuesto, la llamada a esta función devolved la &ue "My funct llll was called• en el navegador.

Las funciones integradas est6n disponibles para su uso en todas las secuend• de comandos de PHP, pero si declara sus propias ~ones, sólo estarin dispon¡. bles para la secuencia 0 secuencias de comandos en las que se declaren. Conviene incluir en un archivo las funciones mú utilizadas y recurrir a un instrucción require (› dentro de las secuencias de comandos para acceder a ellas.

Dentro de una función, las llaves encierran el código que realiza la tarea requeridL En su interior se pueden incluir todos aquellos elementos vilidos de PHP como llamadu de función, declaraciones de nuevas variables o funciones, instrucciones require () o iuclude (l y código HTML. Si queremos salir de PHP dentro de una función y escribir HTML, podemos hacerlo de la misma forma que en cualquier otra parte de la secuencia de comandos: mediante una etiqueta de cierre de PHP seguida del código HMTL A continuación se recoge una modificación vilida del ejemplo anterior que genera el mismo resultado:

```
c?php
    fuactioa ,oy_fuac:tioa()
    {
?>
Ny fuactioa wa• cdled
c?pbp
    )
?>
```

Fijese en que el código de PHP aparece encerrado enm etiquetas de~ y cierre de PHP. En la mayor parte de los ejemplos de c6digo utilizados en este libro

;eñ~ra;:;.::~~ ~?zyirz:'~~~:~~~~!~

Designar funciones

Lo mú importante que recordar a la hora de asignar nombres a las funciones es que deberfan ser cortos y descriptivos. Si su función aea un encabezado de pigina, enca.beaadopag () o enca.bezado_pag serian buenas elecciones.

A continuación se indican varias limitaciones:

- No se puede utilizar el mismo nombre asignado a una función existente.
- El nombre de la función sólo puede contener letras, dlgitos y guiones bajos.
- El nombre de la función no puede comenzar por un dlgito.

Muchos lenguajes permiten volver a utilizar nombres de funciones. Esta opción se conoce como sobrecarga de funciones. Sin embarg~, PHP no admite esta posi~ilidad por lo que sus funciones no pueden tener el DUSDIO nombre que una función incorporada o una ~ción existente definida por~ usuario. Ffjese en que aunq~e las secuencias de comando de PHP conocen las funciones integradas en el lengua,.:,

EL_HOLISTICO@HOTMAIL.COM - 2006

las funciones definidas por el usuario sólo existen en las secuencias de comandos declaradas. Por lo tanto, podemos volver a utilizar el nombre de w,a función en un archivo diferente aunque esta prktica genera confusión y debería evitarse.

Los siguientes nomblft de función son válidos:

```
:i&MCl
name211
:iaae_tllreeCl
_u•fourll
```

Estos otros no lo son:

```
su-o
::aae-aiaCl
!opeDCl
```

(El último seria v6lido si no existiese una función con el mismo nombre.)

Aunque $name no es un nombre v61ido de función, una invocación como

```
$name();
```

se ejecutaría correctamente, en función del valor de $name. Se debe a que PHP toma el valor almacenado en $name, busca una función con ese nombre e intenta invocarla. Este tipo de función se denomina función de variable y puede que en ocasiones le resulte útil.

Utilizar parámetros

Para poder realizar stt-trabajo.-la-mayor partNIe.las funciones requieren el uso de uno o varios par6metros. Un par6metro permite pasar datos dentro de una función. A continuación, se recoge el ejemplo de una función que requiere un parámetro. Esta función toma una matriz unidimensional y la muestra en una tabla.

```
~UAction create_table(Sdatal
{
   echo •eta.ble border • 1>':
   re••t(Sdatal,   // Recuerde que •e utiliza para apuntar al inicio
   svalue • current(SdataJ,
   vhile  ($Yalue)
   {
      acbo •etr>etd>$valuee/td>e/tr>\n•:
      S•alue • neat!$datal,

   echo 'e/table>':
```

Si llamamos a la función create_table() de la siguiente forma:

```
s.i, array • array('LiDe —e. •. 'Liu  t—:', 'Liu  three. '>:
cr91Tte_table(Say_array):
```

se generan el resultado ilustrado en la figura 5.4.

FIgunl 5.4. Esta tabl.l HTML • el resultado de llamar a la funclc)n c:reate_table().

Los par6metros nos permiten obtener datos creados fuera de la función (en este caso la matriz $data) dentro de la función. Como ocurre con las funciones incorporadas, las definiciones establecidas por el usuario pueden tener varios par6metros asf como par6metros opcionales. Podemos mejorar la función creace_table () de varias formas, pero una de ellas podrfa consistir en permitir que el invocador especifique el borde de la tabla u otros atributos. A continuación, se recoge una versión mejorada de la función. Resulta muy similar, pero nos permite establec;er la anchura del borde de la tabla asf como el espacio y relleno de Ias ql_das.

```
functiOD create_tabla21 $data, S!lorder •1, Scell~iag • 4, Sc:4S.q,aciag • 4
{
   echo •etüle border•'$1:1order•  cellpaddiDg • •scellpaddiag'~
      .•  cellepaciag•'$cellepaciag'>";
   reeet ISdata) :'
   I't&tu. .•  i:arnnt"tSdn•t:
   vhil•  ISvalue)
   {
      acllo •ttr»etd>Svaluec/td~a,/tr:0\n• :
      Svalue • uat($datal,

   echo •e/tal,la>'1
```

El primer par6metro de crea te_ table2 () sigue siendo obligatorio. Los otros tres par6metros son opcionales porque se han definido valores predeterminados. Podemos obtener un resultado muy similar al ilustrado en la figura 5.4 con la siguiente llamada a create_table2 ().

```
create_table2($ay_array);
```

Si queremos mostrar los mismos datos con mú detalles, podríamos llamar a la función de la siguiente forma:

```
create_table2($ay_array, 3, 1, 11;
```

No es necesario indicar los valores opcionales (podemos suministrar algunos e ignorar otros). Los par6metros se asignar6n de izquierda a derecha.

Recuerde que 11(? se puede dejar sin definir un pamnetro opcional y definir otro situado por detús. En este ejemplo, si desea pasar un valor para cellapacing,

EL_HOLISTICO@HOTMAIL.COM - 2006

tambiffl deber' pasar otro cellpadding. &ta suele ser una fuente común de errores de programación. También es la razón por la que los parámetros opcionales se especifican al final en cualquier lista de parámetros.

La siguiente llamada de función:

```
create_table2($ay_array, 31;
```

resulta perfectamente válida: establece $border en 3 y deja los parámetros $cellpadding y $cellspacing en sus valores predeterminados.

También puede declarar funciones que acepten un número de parámetros variable. Por medio de las funciones de ayuda func_num_args ( )•, func_get_arg () y func_get_args () puede saber cuántos parámetros se han pasado y qué valores tienen. Veamos un ejemplo:

```
function var_arga()
(
   echo o1'1111ber of para..tera:•;
   echo func_nua_arga(I;

   echo •cbr />';
   $arga • func_get_argall;
   foreach 1$arga •• $arg)
      echn $arg.'cbr />•;
) ;
```

Esta función indica el número de parámetros pasados e imprime cada uno de elles.-La funci6n r"URC-_DL1111 ...-&rga( ). devuelve elnúmerode ~ pasados. La función func_get_args () devuelve una matriz de los argumentos. También p.!,ied~ acced~ a_ 1°5. a~~ UI)Q !tuno. por .~ediQ dJ.~-ÑI!d~R. ~w.ic;:_g~L-arg ( ) , a la que debe pasar el número de argumento al que desee acceder. (Los argumentos se .numeran empezando desde cero.)

Ámbito

Es posible que se haya fijado en que cuando necesitamos utilizar variables dentro de un archivo requerido o incluido, basta con declararlo en la secuencia de comandos antes de la instrucción require () o include (), pero al utilizar una función, pasamos dichas variables dentro de la función de manera explícita. Esto se debe en parte a que no existe ningún mecanismo para pasar variables explfcitamente a un archivo requerido o incluido y en parte a que el ámbito de las variables se comporta de forma diferente en las funciones.

El ámbito de una variable controla dónde resulta visible y el lugar en el que se puede utilizar. Cada lenguaje de programación adopta reglas diferentes a la hora de establecer el ámbito de las variables. Las de PHP son bastantes simples:

- El ámbito de las variables declaradas dentro de una función abarca desde la instrucción en la que se han declarado hasta las llaves de cierre situadas al final de la función. tste es el ámbito de función y las variables declaradas asi se denominan variables locales.
- El ámbito de las variables declaradas fuera de funciones abarca desde la instrucción en la que ~ declaran hasta el final del archivo, pero no en el interior de funciones. Este ámbito es el ámbito global y las variables declaradas así se denominan variables globales.
- Las variables superglogales resultan visibles tanto dentro como fuera de las funciones (si desea obtener un listado de estas variables, consulte un capítulo anterior).
- El uso de las instrucciones require () e include () no afecta al ámbito. Si la instrucción se utiliza dentro de una función, se aplicará el ámbito de la función. Si no se encuentra dentro de una función, se aplica el ámbito global.
- La palabra clave global se puede utilizar manualmente para especificar que una variable definida o utilizada dentro de una función tiene ámbito global.
- Las variables se pueden eliminar manualmente llamando a U;D••t ($nombre_variable). Una variable deja de tener ámbito si se elimina.

Los siguientes ejemplos le ayudarán a aclarar estos conceptos. El fragmento de código que se incluye a continuación no genera ningún resultado. En concreto, se declara una variable llamada $var dentro de la función fn (). Como la variable se declara dentro de una función, suámbito es el de una función y sólo exiatirá desde donde se declara hasta el final de la función. Cuando volvamos a hacer referencia a $var fuera de la función, se creará una nueva función llamada $var. Esta nueva variable tiene ámbito global y estará visible has~-ef fiñal del archlvo~Por desgracia, si la única instrucción que utilizamos con la nueva variable $vares echo, nunca tendrá un valor.

```
function fn II
{
   $var • •contencs•;

echo $var;
```

El siguiente ejemplo muestra lo contrario. En él se declara una variable fuera de la función y se intenta utilizar dentro de otra.

```
function fn II
{
   echo 'inaide the function, $var • '.svar.'cbr    />';
   $var • •contenc•2';
   echo •inaide the function, $var • '.$var.'cbr    1~1

$var • 'content• 1';
fnll;
echo •outaide the function, $var • '.$var.'cbr    />'1
```


EL_HOLISTICO@HOTMAIL.COM - 2006

El resultado de este código será el siguiente:

```
inaid• the function, Svar •
inaide the function, Svar • content• 2
outeide the function, Svar • cantent• 1
```

Las funciones no se ejecutan huta que no se llaman, por lo que la primera instrucción es $var • 'contenta 1' :. &ta crea una variable llamada $var, con ámbito global y su contenido es •con tents 1•. La siguiente instrucción ejecutada es una llamada a la función fn (). Las lineas incluidas dentro de la instrucción se ejecutan en orden. La primera línea de la función hace referencia a la variable llamada $var. Al ejecutar esta lfnea, no se puede ver la v"ariable $var creada anteriormente, por lo que crea una nueva con ámbito de función e imprime si(valor. Esto crea la primera linea de los resultados.

La siguiente línea de la función establece el contenido de $varen •contenta 2•. Como estamos dentro de la función, esta linea cambia el valor de la variable $var local, no de la global. La segunda línea del resultado verifica que este cambio funciona.

Con esto llegamos al final de la función y se ejecuta la secuencia de comandos. La instrucción echo demuestra que el valor de la variable global no ha cambiado.

Si queremos que una variable creada dentro de una función tenga ámbito global, podemos utilizar la palabra clave global de la siguiente forma:

```
function fn(I
(
   9lobal svar1
   tvar • 'conc•nta•;
   e.cho •in•ida th• function. $var. '.$var.'cbr />';

fnO;
echo •out•ide th• function, Svar • •.svar.'cbr />';
```

En el siguiente ejemplo, la variable $var se ha definido como global lo que significa que tras llamar a la función, la variable también saldrá fuera de la función. El resultado de esta secuencia de comandos será el siguiente:

```
inaida tha functian, $var • contant•
out•ide the function, $ver• contante
```

El ámbito de la variable abarca desde el punto en el que se ejecuta la linea ~lobal $var;. Podñamos haber declarado la función por endma o debajo de la lana de llamada. (Como observará, existe bastante düerencia entre el únbito de función y el ámbito de variable.) La ubicación de la declaración de función no tiene re~vancia. Lo importante es el lugar en el que se llame a la función y, por lo tanto, dade el que se ejecuta el c6digo incluido en su interior. También puede utilizar la palabra daveglobal en la parte superior de una secuencia de comandos al utilizar por primera vez una variable para declarar que su ámbito abarca la secuencia de coaumdos. &te es probablemente el uso más comdn de la palabra clave global.

Como puede ver por los ejemplos anteriores, resulta perfectamente válido volver a utilizar un nombre de variable para una variable dentro y fuera de una función sin que se produzcan interferencias entre ambas. Sin embargo, no conviene utilizar el mismo nombre porque sin un examen atento del c6digo y del ámbito podría pensarse que se trata de las mismas variables.

Llamadas por referencia frente a llamadas por valor

Si desea escribir una función llamada incre—nt () que nos permita incrementar un valor, podríamos vemos tentados a escribir lo siguiente:

```
function incra..at($valua, $amount • 1l
(
   $valua • Svalue +$amount1
```

Este código no sirve de nada. El resultado del siguiente c6digo de prueba será 10.

```
Svalu11 • 10;
incr —— t (Svalua);
echo $valu;
```

tós c:oñlenidoi dé $váliie rié> han ciunbladodebtdoaIu reglu dl!'bdrito;Este c6digo crea una variable llamada $value que contiene 10. A continuatión llama a la iunción increment O. Se l:rea.la vJ1riable$:v•lue die la función al 1—1¡l. Se le agrega una unidad, por lo que el valor de $value pasa a 11 dentro de la función hasta qu~ llega a su fin y volvemos al c6digo que la llamó. En este c6digo, la variable $value es diferente porque tiene ámbito global y no varia. Una forma de resolver este problema consiste en declarar $val ue en la función como global, pero esto implica que para poder utilizarla, la variable que queremos incrementar deberá llamarse $value. Una solución mejor seria utilizar las llamadas por referencia.

La forma normal de llamar a los parámetros de una función es pasándolos por valor. Al pasar un parámetro, se creará una nueva variable que contiene el valor pasado. Se trata de una copia del original. Puede modificar este valor como desee, pero el valor original de la variable fuera de la fundón seguirá siendo el mismo.

Una solución mejor es utilizar las llamadas por referenda. En este caso, cuando se pasa un parámetro a una función, en lugar de aear una nueva variable, la función recibe una referencia a la variable original. Esta referencia tiene un nombre de variable, que comienza por el símbolo del dólar($) y se puede utilizar de la misma forma que cualquier otra variable. La diferencia estA en que en lugar de tener un valor propio, simplemente hace referencia al original. Cualquier cambio realizado a la referencia sólo afectará al original.

Para indicar que un parámetro debe pasarse por referencia se coloca un símbolo & delante del nombre del parúnetrO en la definidón de función. No es necesario

EL_HOLISTICO@HOTMAIL.COM - 2006

ningún cambio en la llamada de función. El ejemplo de función anterior, increment ( ), se puede modificar para pasar un parúnetro por referencia y lograr que funcione correctamente.

```
funccion incre .. ntl•Svalue, $amount• 11
(
  svalue • $value +$amaunt;
```

Ahora ya disponemos de una función operativa y podemos utilizar el nombre que deseemos para incrementar el valor que nos plazcL Como se indicó anteriormente, resulta confuso utilizar el mismo nombre fuera y dentrd de una función, por lo que aplicaremos un nuevo nombre a la variable en la secuencia de comandos principal. El siguiente código de prueba imprimir4 10 antes de llamar a increment () y 11 despu&.

```
•• • 10,
ecbo $a.'cbr />';
incre•ent l$•ll
ecbo $a.•cbr />'l
```

Devolver desde funciones

La palabra clave return detiene la ejecución de una función. Cuando una función termina porque todas sus instrucciones se han ejecutado o porque se-ha utilizado la palabra clave return, la ejecución regresa a la llamada de la función.

--Si llama-a la siguiente función, sólo.se..ejecutarála pmnera instrucción.echo..

```
funccion ceac_returnll
(
  ecbo •Thia atat-nt vill be encucad';
  recura,
  ecbo 'Thia atac-nt will nevar be eucutad•;
```

Obviamente, no se trata de una forma muy útil de utilizar la instrucción return. Por regla general, sólo querremos volver desde el centro de una función en respuesta a una condición.

Por ejemplo, una razón habitual para utilizar una instrucción return es una condición de error que detenga una función al final. Por ejemplo, si escribimos una función para determinar el número mú grande de dos especificados, podemos salir de la función si falta alguno de los números.

```
faacUon larger I SX, $y I
(
  U l"lieHCl$xllllieHtl$y))'
  (
    ec!1o 'Thia functian require• t- mmbera•,
    return,
```



```
|
U ($X>•$y)
  acbo $X;
el••
  acbo $y¡
ecbo 'cbr />',
```

La función incorporada iaaet ( ) indica si se ha creado una variable y se le ha asi~o .un valor. En este código, devolvemos un mensaje de error y salimos de la función 11 uno de los parúnetros no lleva definido un valor. Para probarla, utillzamm ! isset (l, que significa "NO iHet ()•• de manera que la instruc:ción if pueda leerse como "si x no est4 establecida o si y no est4 establecida". La función devolverá.un resultado si cualquiera de las dos condiciones resulta ser cierta.

Si se ejecuta la instrucción return. se ignorarán las siguientes líneas de la función. La ejecución del programa regresad al punto en el que se invocó la función. Si se establecen ambos par4metros, la función imprimirá.el mayor de los dos•

El resultado del siguiente código:

```
$a• 11
Sb. 2.s,
Se• 1.91
laqerC$a, $bl;
laqerl$c, $al;
lar9erl$d, Sal;
```

ser4

```
2.5
1.9
```

--Tiiá timé:üon°require•°-l:wci-nuiíb.u

Devolver valores desde funciones

La salida de una función no es la única razón para utilizar la instrucción return. Muchas funciones utilizan instrucciones return para comunicarse con el código que las llamó. En lugar de imprimir los resultados de la comparación de la función lai:ger (l, puede que resulte mú útil devolver la respuesta. De esta forma. el códi~ ~e llame a la función puede decidir si mostrar o utilizarla y cómo. La función u,corporada auaxl1 es equivalente y se comporta de esta forma.

Podemos escribir la función larger () así:

```
function larver ISz. $yl
(
  if (liaaeCl$xIIIliaaecl$yll
    recura faln1
  elae H 1$X>•$y)
    ncurn $x;
  •l•e
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
    retum  Sy,
```

En este código devolvemos el valoi: mayor de los dos pasados. Obviamente, el valor devuelto será diferente en caso de error. Si uno de los números desaparece, devolveremos false. El único pero de este enfoque es que los programadores que llamen a esta funáón deben probar el tipo de devolución con ••• para asegurarse de que false no se confunde con 0. La función max (1 de PHP no devuelve nada si se establecen ambu variables y si sólo se establece una, será la que devolverá.

El siguiente código:

```
Sa • 11  Sb • 2.5;   Se• 1.t1
echo larger(Se, SbJ.•cbr  />";
ecbo largerCSc, SaJ."cbr  />"1
ecbo  laz,Jerl$4, SaJ.•cbr  />";
```

generará el siguiente resultado porque $d no existe y false no está visible:

```
2.5
1.t
```

Las funáones que realizan alguna tarea, pero que no necesitan devolver un valor, a menudo devuelven true o falae para indicar si se realizan satisfactoria-D\ente o fallan. Los valores booleanos true y falae se pueden representar con válores 1 y 0, respectivamente, aunque son de tipos diferente.

Bloques de código

Pua ciec:iarÜ que un giupo de iiüitrüéaoñes fórma-wi"&Toque l.ü toloc:amos entre llaves. Esta estructura no tendrá efectos sobre el funcionamiento del código pero sf implicaciones especfficu como la forma en la que se ejecutan las estructuras de control, como los bucles y las estructuras condiáonales. A continuación se incluyen dos ejemplos que funáonan de forma muy diferente:

Ejemplo sin bloque de c:6_digo

```
forl$1 • 01  $1  e 11  $1++ 1
   eello •Line lcbr />';
ecbo  'Li-  2cbr />' 1
```

Ejemplo con bloque de código

```
forCSi • 01  Si  e 11  Si++ 1
(
   ecbo 'Line  lcbr />';
   ec:Jao 'Line 2cbr />'1
```

En ~ ejemplos, el bucle for se itera tres veces. En el primer ejemplo, el bucle for sólo ejecuta la lfnea situada inmediatamente por debajo. El resultado de este ejemplo es el siguiente:


```
Line 1
Line 1
Line 1
Line 2
```

El segundo ejemplo utiliza un bloque de código para agrupar las dos lineas. Esto implica que el bucle for ejecuta ambas lfneas tres veces.

```
Line 1
Line 2
Line 1
Line 2
Line 1
Line 2
```

Como el código de estos ejemplos está sangrado correctamente, es probable que vea la diferencia enseguida. El sangrado del código tiene el objetivo de ayudar a los lectores a interpretar visualmente lu Uneu afectadas por el bucle for. Sin embargo, ffjese en que los espaáos no afectan a la forma en la que PHP procesa el código.

En algunos lenguajes, los bloques de código afectan al ámbito de las variables. &te no es el caso de PHP.

Implementar recursión

PHPa4mite~de .t:eCUAióA Las.fuadoaes de r:ena:sióo•«ibman a sí mismas. Estas funciones resultan espeáalmente útiles para desplazane l l Pr estructuras de datos dinámico como listu y árboles vinculados.

$iiúmibai:go;son eic:uas las aplicaaoñes biadás'eida'Wetiquériqüieran liria-- estructura de datos de esta complejidad por lo que el uso de esta funáón es limitado. La recursión se puede utilizar en lugar de la iteraáón en muchos casos porque ambas permiten realizar una operacidn de manera repetida. Las funáones recursivas resultan mú lentu y utilizan IIIÚ memoria que la iteración. razón por la cual debería dar prioridad a estás últimu.

En el listado 5.5 se recoge un breve ejemplo de fÚnciónrecuniva.

Listado 5.5. IIICUnllon.php. Invertir una CIIdena ulllzllndo la rec:unil6n y la ileracl6n.

```
functiOD rev.ree_rC$•trl
(
   if CetrlenCS•trl>OI
      rever••-rC•u.betrCSetr, 1111
   ecbo irab•tr(Setr, 0, 111
   returnl


functiOD rever••-il$•trl
(
   for 1$1•11 Sic••trl:inCS•trl; Si••I
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
    echo aub•tr($atr,  -Si,  11;
  retum;
```

En este listado, hemos implementado dos funciones. knbas imprimen una cadena al revés. La función reverae_r () es recursiva y la función reverae_i () es iterativa.

La función reverae_r () toma una cadena como parámetro. Al llamarla, se llamará a sí misma y pasará el penúltimo carácter de la cadena. Por ejemplo, si llama a

```
rever•e_rl'Hello'll
```

9e llamar6 a sf misma varias veces con los siguientes parámetros:

```
rever••  r('ello'l1
rever••- rl •110• 1l
rever•e-rl•10•)1
rever•• -rl •o• 1,
rever••=r 1' 'l ı
```

Cada llamada que realiza a sí misma crea una nueva copia del código de funci6n en la memoria del servidor pero con un parámetro diferente. Es como si pretendi4ramos llamar a una función diferente cada vez. Esto evita que las instancias de la función se confundan.

En cada llamada, se prueba la longitud de la cadena pasada. Al llegar al final de la cadena (strlen(>••0) falla la condición. La instancia mú reciente de la funci6ft'(reverse;_rt'.a-,t-seguirá adelante y-ejecutan la siguien~lfnea-de-código que debe imprimir el primer carácter pasado de la cadena (en este caso, no ~y ningdn carácter porque la cadena está vada).

A continuación, esta instancia de la función devuelve el control a la instancia que la ~. en concreto reverae_r(•o'). Esta instrucción imprime el primer carácter eruu cadena (•o•) y devuelve el control a la instancia que lo llamó.

El proceso continúa (se imprime un carácter y se vuelve a la instancia de la función superior en el orden de llamada) hasta que el control vuelve al programa principal.

Las so•uciones recursivas presentan un aspecto muy elegante y matemático. En la mayor parte de los casos, sin embargo, resulta mucho mejor utilizar una solución iterativa. El listado 5.5 también incluye la variante iterativa. Como observará, es igual de larga (aunque no siempre es así) y realiza la misma funci6n. La diferencia principal es que la función recursiva realiza copias de sí misma en memoria, lo que implica la sobrecarga de varias llamadas de función.

La versión recwsiva se puede utilizar cuando el código resulta mú breve y elegante que el de la versión iterativa, cosa que en este dominio de aplicaciones no suele ocurrir a menudo. Aunque la recursión parece mú elegante, los programadores ~úelen olvidarse de incluir una condición de finalización, lo cual implica que la

función se repetirá hasta que el servidor se quede sin memoria o hasta que se exceda el tiempo de ejecución.

Lecturas adicionales

El uso de las funciones include (), require (>, function y return tambh!n se explica en el manual en lútea. Si desea saber mú sobre conceptos como el de la recursi6n, las llamadas por valor o por referencia y el ámbito de las variables que afecta a muchos lenguajes, consulte un libro de texto general sobre infonmtica.

A continuación

Ahora que ya sabemos utilizar los archivos de inclusión, los archivos n=queridos y las funciones para lograr que nuestro código resulte mú sencillo de man~r y susceptible de utilizar de manera repetida, en el siguiente capítulo se analizañ el software orientado a objetos y la compatibilidad que ofrece PHP al respecto. El uso de objetos permite lograr objetivos similares a los conceptos presentados en este capítulo pero con ventaju superiores para proyectos complejos.

EL_HOLISTICO@HOTMAIL.COM - 2006

6

PHP orientado a objetos

En este capítulo se explican conceptos relacionados con el desarrollo orientado a objetos y se muestra cómo implementarlos en PHP. PHP S presenta w,a nueva y más completa implementación orientada a objetos que aumenta la utilidad de clases y objetos.

Ahora la implementación orientada a objetos de PHP dispone de todas las funciones que se esperan de un lenguaje completamente o~entado a objetos. Analizaiemos cada una de estas caracteristicas según avancemos por el capítulo, en especial para los que hayan utilizado PHP 4 o una versión anterior.

Entre los temas clave que trataremos en este capítulo se incluyen los siguientes:

- Conceptos orientados a objetos
- Clases, atributos y operaciones
- Atributos de dase
- Constantes de dase
- Llamadas a métodos de clase
- Herencia
- Modificadores de acceso
- Métodos ~ticos
- Tipos

EL_HOLISTICO@HOTMAIL.COM - 2006

- Clonar objetos
- Cases abstractas
- Diseilar clases
- Implementardiseños personalizados
- Funciones orientadas a objetos avanzadas
- Nuevas funciones orientadas a objetos de PHPS

Conceptos orientados a objetos

Los lenguajes de programación modernos suelen admitir o incluso necesitan adoptar un enfoque orientado a objetos para el desarrollo de software. El desarrollo orientado a objetos (00) intenta utilizar las clasificaciones, las relaciones y las propiedades de los objetos del sistema para contribuir al desarrollo de programas.

Clases y objetos

En el contexto del software orientado a objetos, un objeto puede ser pñctica-mente cualquier elemento o concepto (un objeto físico como una mesa o un cliente; o un objeto conceptual que sólo existe en el software, como un campo de entrada de texto o un archivo). Por regla general, estaremos mú interesados en objetos conceptuales, incluidos objetos del mundo real que necesitan representarse en soft· W!lff,

El software orientado a objetos se disetla y construye como un conjunto de objetos independientes con atributos y operaciones que interactúan para cubrir nuestras necesidades. Los atributos son propiedades o variables relacionadas con el objeto. Las operaciones son métodos, acciones o funciones que el objeto realiza para modificarse a sí mismo o para obtener algún efecto externo. (E tffmino atribu· to se utiliza en lugar de los términos variable miembro y propiedad, y el tfrmino operación como sinónimo de método.)

La ventaja principal del software orientado a objetos es su capacidad para admi- tir o fomentar la encapsulación, tambim conocida como ocultación de datos. B6si-camente, el acceso a los datos dentro de un objeto sólo est4 disponible a trav4!s de operaciones con objetos, conocida como la interfaz del objeto.

La funcionalidad de un objeto está vinculada a los datos que utiliza. Podemos alterar los detalles de la implementación del objeto para mejorar el rendimiento, agregar nuevas funciones o resolver errores sin tener que modificar la interfaz, cuyos efectos pueden extenderse a todo el proyecto.

En otras áreas del desarrollo de software, la programación orientada a objetos es la nonna *y* el software orientado a funciones se considera desfasado. Por varias razones, la mayor parte de las secuencias de comandos orientadas a objetos sigue;

por desgracia, disetlada y escrita utilizando una metodología orientada a fundo. nes. Existen varias razones que explican este enfoque. La mayoría de los proyectos Web son relativamente pequeños y sencillos. En la mayor parte de los casos, los proyectos Web° se pueden desarrollar satisfactoriamente sin planificar el enfoque debido a su tamai\o. Sin embargo, si intenta aplicar la misma Iffllica para construir algo de mayor tamailo, la calidad de los resultados se ver, afectada, si es que se obtiene algún resultado. Lo mismo sucede con proyectos de software de mayor tamafto.

Muchos proyectos evolucionan a partir de un conjunto de páginas vinculadas para convertirse en una aplicación compleja. Estas aplicaclones, ya se presenten a trav4!s de cuadros de dWogo y ventanas o páginas HI'ML generadas dinámicamente, necesitan disponer de una metodología de desarrollo correctamente ideada. La orientación a objetos puede ayudarle a gestionar la complejidad de los proyectos, a inaementar la reusabilidad del código y, por tanto, a reducir los costes de manteni-iniento.

En el software orientado a objetos, un objeto es una colección muca e identifica-ble de datos y operaciones almacenados que operan sobre dichos datos. Por —- plo, podemos tener dos objetos que representen botones. Aunque •mbos tengan el texto •Aceptar", una anchura de 60 pfxeles, una altura de 20 píxeles y otros atribu-tos idmtlcos, necesitamos disponer de la posibilldad de trabajar con uno o con el otro. En software, existen variables independientes que actúan como identificadores únicos para dichos objetos.

Los objetos se pueden agrupar en clases. Las clases representan un conjunto de objetos que puede variar, pero que deben tener un conjunto de características en común. Una clase contiene objetos con operaciones que se comportan de la misma fonna_y. atributos que repn!lelltan las mislna$cosas, aunque los-valores-de-dichos--- atributos varíen de objeto a objeto.

Podemos imaginar el término bicicleta como una clase de objetos que describe muchas bicicletas diferentes con una gran cantidad de características o atributos en común; por ejemplo, que tengan dos ruedas, un color y un tamaño, y realicen opera-dones, como moverse.

Mi bicicleta es un ejemplo de objeto que encaja en la clase bicicleta Compartid rasgos comunes a todas las bicicletas, incluida una operación de movimiento que comporta de la misma forma en la mayoria de las bicicletas (aunque apenas la utilice). Mi bicicleta tiene atributos con valores mucos porque es verde y no todas las bicicletas tienen ese color.

Polimorfismo

Un lenguaje de programacim orientado a objetos debe admitir el polimorfismo, lo que significa que las clases pueden tener diferentes comportamientos con respec-to a la misma operación. Por ejemplo, si tenemos una clase coche y una clase bici· cleta, sus operaciones de desplazamiento serú l diferentes. Para los objetos del mundo real, este hecho no plantea un problema ya que es poco probable que se confunda el

EL_HOLISTICO@HOTMAIL.COM - 2006

movimiento de una bicicleta con el de un coche. Sin embargo, ún lenguaje de programación no posee el sentido común del mundo real, por lo que el lenguaje debe admitir el polimorfismo para determinar qué operadón de movimiento utilizar con un objeto dado. El polimorfismo es más una característica relacionada con los comportamientos que con los objetos. En PHP, sólo las funciones miembro de una clase puede ser polimórficas. Una comparación del mundo real es la de los verbos en los lenguajes naturales, que equivalen a las funciones miembro. Considere las formas en las que una bicicleta se puede utilizar en el mundo real. Se puede limpiar, mover, desmontar, reparar o pintar, entre otru cosas.

Estos verbos describen acciones genéricas porque no se puede saber el tipo de objeto sobre el que se actúa. (Este tipo de abstracdón sobre objetos y acciones es una de las características que distinguen a la inteligenda hu.mana.)

Por ejemplo, la operación de mover una bicicleta requiere acciones completamente diferentes a las necesarias para mover un coche, aunque los conceptos sean similares. El verbo mover se puede asociar a un conjunto concreto de acciones una vez que se dé a conocer el objeto sobre el que se actúa.

Herencia

La herencia nos permite crear una relación jerárquica entre clases utilizando subclases. Una subclase hereda atributos y operaciones de su superclase. Por ejemplo, un coche y una bicicleta tienen elementos en común. Podemos utilizar una clue vehícufo pata iñctutrntementosWDKJW, atributo ele eelar y w operación de movimiento común a todos los vehículos y permitir que nuestru clases coche y bicicleta las hereden. Comprobañque los túminos subclase, clase derivada y secundaria se utilizan como sinónimos; lo mismo ocurre con supeiclase 'y' principal: -

Gradas a la herencia, podemos desarrollar y seguir agregando elementos a clases existentes. A partir de una clase búica se pueden derivar clases mú complejas y especializadas a medida que surja la necesidad. Como resultado, nuestro código será más reutilizable, lo cual es una de las ventajas más importantes de un enfoque orientado a objetos.

El uso de la herencia puede ahorramos trabajo si las operaciones se escriben una vez en una superclase en lugar de varias veces en subclases diferentes. También puede ayudarlos a modelar de manera más exacta relaciones del mundo real. Si se establece una relación lógica del tipo ·es un/una" en una frase entre dos clases, es probable que la herencia resulte adecuada. La frase "un coche es un vehfculo" tiene sentido, pero la frase ·un vehículo es un coche" no lo tiene porque no todos los vehículos son coches. Por lo tanto, el coche puede derivarse desde la clase veh1culo

Crear clases, atributos y operaciones en PHP

Por el momento, las clases se han estudiado de forma bastante abstracta. Al crear una clase en PHP, " utiliza la palabra clave class.

Estructura de una clase

La definición búica de una clase presenta este aspecto:

```
el••• DOllbreela••
{
}
```

Para que la clase resulte útil necesita tener atributos y operaciones. Para crear atributos se declaran variables dentro de una definición de clase utilizando la palabra clave var. El siguiente código crea una clase llamada classname, con dos atributos $attributel y $attribute2.

```
class classname
{
   var Sattributel;
   var Sattr1but•21
```

Las operaciones se crean declarando funciones dentro de la definición de clase. El siguiente código crea una clase denominada classname con dos operaciones que no hacen nada. La operación operationl () no toma parámetros y la operación operation2 () toma dos pañmetros.

```
el••• el•••-
{
   f.1111ct!on operationl(l
   (
   |
   tlllle.tion ope:ratioD2j sea:r,•J., ~parl!_II?l.. -
   {
   |
```

Constructores

La mayor parte de las clases constan de un tipo especial de oper&l=iónllamado constructor. Un constructor se llama al crear un objeto y suele realizar tareas útiles de inicialización como establecer atributos con valores de inicio lógicos o crear otros objetos necesarios.

Un constructor se declara de la misma forma que otras operaciones, pero tiene el nombre especial _ construct (). Es uno de los cambios de PHPS. En versión anteriores, las funciones de constructor IeJÚaJ\ el mismo nombre que la clase. Por motivos de compatibilidad inversa, si en una clase no se encuentra una función con el nombre_ construct ( ), PHP bu,scaráuna función que tenga el mismo noinbre que la clase.

Aunque podemos llamar manualmente al constructor, su fund6n principal es hacerlo automtica~te al crear un objeto. El siguiente código declara una clase con un constructor:

EL_HOLISTICO@HOTMAIL.COM - 2006

```
el••• el•••na-
(
   func:tion _ cOJ1at:ruct(Sparaa)
   (
      aeho •conat:ruetor called with para..ter Sparaa cbr />";
```

PHPS admite actualmente la sobrecarga de funciones, lo que significa que puede proporcionar una función con el mismo nombre y diferentes tipos o cantidad de parámetros. (Muchos lenguajes de programaciónorientadosa objetos lo admiten.) Lo analizaremos mú adelante.

Destructores

Lo contrario de un constructores un destructor. Los destructores son una de las novedades de PHPS. Le permiten ejecutar una funcionalidad concreta antes de que se destruya una clase, lo que ocurrirá automáticamente cuando se anulen todas las referendas a la clase o éstas no se correspondanal ámbito.

Aligual que ocurre con los nombres de los constructores, es necesario asignar _ destruct () al nombre de un destructor de clases. Los destructores no pueden adoptar parámetros.

Crear instancias de clases

- Tras declarar una clase, tenemos que crean11robjeto (un elemento concreto que forme parte de una clase) con el que trabajar. Esta tarea también se conoce como crear una instanda o instanciar una clase. Para crear un objeto se utiliza la palabra clave new. Es necesario especificar a qué clase pe~neceráel objeto y suministrar todos los parámetros que requiera el constructor.

El siguiente código declara una clase llamada classname con un constructor y crea tres objetos del tipo classname:

```
el••• cla••na-
(
   funetion _conat:ruct(Spara•I
   (
      echo •conatruc:tor callad with parameter Sparaa cbr />";


Sa • new elaa~ ( ' Pirat•);
Sb • new claaanaae('Second');
Se• a.- claa.....();
```

Como el constructor se llama cada vez que creamos un objeto, este código genera el sigui~te resultado:


```
Conat:ructor called with paraaatar Pirat
Conat:ruccor called with paraaater Sacond
Conat:ructor callad with paraaeter
```

Utilizar atributos de clase

Dentro de una clase, disponemos de acceso a una variable especial llamada $this. Si un atributo de la clase actual se denominara $attribute,se utilizaría la secuencia $this- >&ttribute para hacer referencia a él al establecer o acceder a la variable desde una operación dentro de una clase.

El siguiente código muestra cómo establecer o acceder a un atributo dentro de una clase:

```
el••• claaaname
(
   var $attributa;
   funccion operationlSparaa)
   (
      $thia->attribuca • $para•
      echo $this->attribute1
```

La posibilidad de acceder a un atributo desde el exterior de una clase se determina mediante modificares de acceso, que analizaremos 111'5 adelante. En este ejemplo no se restringe el acceso a los atributos, por lo que puede acceder a ellos desde el exterior de clase, como se indica a continuación:

```
el••• cluanaaa
(
   var Sattributa,
I
$a. new cluan...()1
$a->attributa • 'valua';
echo $a->attribute;
```

No es buena idea acceder directamente a los atributos desde fuera de una clase. Una de las ventajas de un enfoque orientado a objetos es que (omenta la encapsulación, lo que puedeconseguir por medio de las funciones_get y_ set. Si en lugar de acceder a los atributos de una clase directamente,se escriben fundones de descriptores de acceso, todos los accesos se realizarán a través de una única sección de código. Las funciones de descriptores de acceso pueden presentar el siguiente aspecto al principio:

```
clua claaanaaa
(
   var $attribute;
   func:tion __9at($n,aae)
   (
```

```
    return $this->$name;
}
functiOD _ ..t l$11a11e, $value1
{
    $thi• ·>$n&.. • $value1
}
```

Este código proporciona func~ones para acceder al atributo denominado $at tribute. Tenemos una función llamada __get (› que devuelve simplemente el valor de $attribute y una función llamada __set () que asigna un valor nuevo a $attribute.

Apreciará que __gec (› adopta un paúmetro, el nombre de un atributo, y devuelve el valor de dicho atributo. Por su parte, la función _sec () adopta dos parámetros: el nombre de un atributo y el valor en que deseemos establecerlo.

No es necesario invocar directamente estas funciones. El doble guión bajo que aparece delante del nombre indica que tienen un significado especial en PHP, como sucede con las funciones __construct () y __clestruct ().

Se preguntará cómo funcionan. Si aea la instancia de la clase

```
$& • nev cla•enaa&ll:
```

puede utilizar las funciones __get () y __set (l para comprobar y establecer el valor de cualquier atributo.

Si iatl'educe

```
$a•>$attribute • s,
```

—estainstrucción invoca implfcitamente la fwlci6n—set ( ) con elyalor de t.na-. establecido en ·attribuee· y el valor de $value en s. Tendr4 que escribir la función __set () para que realice la comprobación de errores necesa~a.

La función __get (› funciona de forma similar. Si en su código hace referencia a $a->$attribute, esta instrucción invocar, explfcitamente la función __get () para devolver el valor.

A primera vista, podrfa parecer que este código no agrega mucho valor. En su forma actual, es probablemente cierto, pero la razón para suministrar las funciones de descripción de acceso es sencilla: sólo tendremos una sección de código que accede a dicho atributo.

Con un solo punto de acceso, podemos implementar elementos de comprobación para aseguramos de que sólo se almacenan datos pertinentes. Si en un momento posterior necesitamos que el valor de $attribute esti entre cero y den, bastar4 con agregar unas pocu lineas de código y verificar antes de permitir los cambios. Podemos modificar la función __set () para que presente este aspecto:

```
hDCCt,pÍI __HtC$11ev_Taluel
(
   lfC ..... 'attritr~t•'  •• Svalue >•o••  $value <• 100 l
      •Clll•·,..ttril:mate•  $value:
```


Con un solo punto de acceso; podemos cambiar libremente la implementación subyacente. Si por alguna razón, optamos por cambiar la forma en la que se almacena $attribute, las funciones de descriptor de acceso nos permitirán hacerlo variando el código en un solo punto.

Puede que en lugar de almacenar $at tribute como variable queramos recuperarla únicamente desde la base de datos cuando resulte necesario, calcular un valor actualizado cada vez que se solicite, inferir un valor de los valores de otros atributos o codificar los datos con un tipo de dato mú pej¡uefto. Sea cual sea el cambio que decidamos realizar, bastan con modificar nuestras funciones de descriptor de acceso. El resto de las secciones de código no se ver4n afectadas mientras las funciones de descriptor de acceso sigan aceptando o devolviendo los datos que esperan otras partes del programa.

Controlar el acceso con private y public

PHP 5 incorpora nuevos modificadores de acceso que controlan la visibilidad de atributos y métodos, y que se aitaden por delante de las declaraciones de atributos y mitodos. PHP 5 admite los tres siguientes modificadores de accesqr

- La opción predeterminada es publ ic, lo que significa q~ si no especifica un modificar de acceso para un atributo o un mitodo, será public. A los elementos públicos se puede acceder desde dentro o fuera de la clase.
- El modificador de acceso pri vate indica que al elemento mAfcado como tal sólo se puede acceder desde el interior de la clase. Puede utilizarlo en todos los atributos si no utiliza ..;.....9et (Ly __setÜ.. Tain.biéq p~~-opta~.~L. convertir en privados algunos mtodos si se trata de funciones que se utilizarán sólo dentro de la clase. Los elementos privados no se heredan (como veremos más adelante).
- El modificador de acceso protected implica que al elemento sólo se puede acceder desde dentro de la clase. Tambiin se utiliza en subclases. como veremos más adelante. Por el momento, piense en protected como algo intermedio entre publ ic y pri vate.

Para añadir modificadores de acceso a la clase del ejemplo, puede modificar el código de esta forma:

```
class classname
{
   public $attrlbute1
   public function _9et($11&1M11
   (
      return $t~ ·>$na..,
   }
   public function _ ••t l$name, $valuel
   (
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
Scbi•->S ..... • Svalua,
```

Llamar operaciones de clase

Podemos llamar a operaciones de clase de la misma forma que llamamos a los atributos de clase. Si tenemos la siguiente due:

```
class classname
{
   fuaceioa oparacioa11)
   {
   fuaccioa oparacioa21Sparaa1,  SparaazJ
   (
```

y aeamos un objeto del tipo claaaname llamado $a de la siguiente forma:

```
S• • ilaw claamaama0I,
```

Seguidamente Uaanamoa a las operaciones de la misma forma queea otras fund~nes: por su nombre *y* con todos los parámetros que se necesiten entre parintesia. Como estas operadones pertenecen a un objeto en lugar de a funciones normales, tenemos que especificar a qué objeto pertenecen. El nombre del-objeto se utiliza de la misma forma que los atributos de objeto:

```
$•-1!.0P.U:•tioa1l1
Sa·>oparacioa2(12,  •ca•c'l;
```

Si nuestras operaáones devuelven algo, podemos capturar los datos devueltos de la siguiente forma:

```
Sx • Sa·>oparacioa1111
Sy • sa-:,,oparacioa2112, 'ca•C'l;
```

Implementar la herencia en PHP

Si nuestra clue es una subclase de otra, podemos utilizar la palabra clave extenc:l• para especificarlo.

El siguiente código aea una clase llamada B que se deriva de la due llamada A previamente definida.

```
cla .. • ..Caad•    A
(
   var Saccrilluca21
```



```
fuaccioa operacioa211
(
```

Si la clase A se declara de la siguiente forma:

```
el••• A
(
   Yar saccribuca11
   funccioa oparacioal 11
   (
```

todos los accesos siguientes a las operaciones *y* atributos de un objeto de tipo a serian válidos:

```
Sii. -    an,
Sb·:,,oparacioa111;
Sb->aCCribuCa1  • 10;
Sb·:,,oparacioa2C1;
Sb•>aCcrillucaz  • 101
```

Ffjese en que como la dase B amplia la clase A, podemos hacer referencia a operationl (> *y* $attributel, aunque se declararon en la daseA. Como subclase de A, a tiene la misma funcionalidad *y* los mismos datos. Además, a ha declarado un atributo *y* una operaáón propios.

Es importante observar que la hemu:ia sólo funciona en una c:lina:ión. La •ubclue hereda lu funciones de la dase prinápal o superclase, pero éstt no toma ninguna funáón-de aquélla. Por lo tanto, las dos siguientes lineas decódigóiériaii" erróneas:

```
Sa • aaw A01
Sa·>oparacioa1(11
Sa•>aCcribuCal • 10,
Sa•:,,oparaciOD2()1
Sa•>accribucaz • 10,
```

La due A no tiene una operadón operation2 () ni un atributo attribute2.

Controlar la visibilidad a través de la herencia por medio de private y protected

Puede utilizar los modificadores de acceso pri vate *y* protected para controlar qué se hereda. Si especifica un mftodo o atributo como private, no se heredad. Si se especifica como protected, no se podrá ver fuera de la due (como si faera un elemento private) pero se heredará. Veamos el siguiente ejemplo:

```
c7pbp
cla8a  A
(
```

```
pri••t• f'IIIICtioa aperatiOD1()
{
   ecbo •operatiODl called•1

pncected    fUDCtioa operatiaa2U
{
   ec:!IO •apentiaa2 ealllld•1

publie f1111etioa operatioall)
{
   aebo •oparatiaal ealled•1


el••• a axtalldll A
{
   fullc:tioa __coaatruetll
   {
      $thia-~ratiOD1()1
      $thia-~ratiOD2();
      $thia-~ratioall)1

Sb • _. 81
```

Fate cóclígo.aa una opem:i6n.lkadl.Jj¡H>.m JullKA: publ.J,~9~~~~4 y private.8 se hereda de A. En el ~r de 8 puede ínlmtar llamar a la operación de9de la clue principal. La linea

```
$1:ldii~>CipéraUoiai"l)1
```

genera un ~r muy grave, como se indica a continuación:

```
raul  error, cau  to pri•ate Mthod A:aoparationl ()  fram eoatext •a•
```

Este ejemplo demuestra ~ lu operaciones privadas no se pueden invocar desde una clase secundaria.

Si anula el comentario de esta linea. lu dos otru ínvocadones fundonarm. La función protect ed es heredada pero sólo se puede utilizar desde el interior de la clue IN!CUIICWia,c:omohemos hecho aqaí. Sí intenta aftadir la siguiente línea:

```
Sb->aperatiaa2Cl1
```

a la parte final del udüvo, ~ el siguiente error:

```
Fa&al. error, Cllll  to protected -tllall   A•:oparaticm2()  fram c:oatext '
```

Sin embargo. puede invocar operationl ( ) desde el exterior de la clue:

```
Sb->oper•f10DJ()1
```

Todo esto es posible porque se ha declarado *como* public.

Reemplazos

Hemos mostrado una subclase que declara nuevos atributae y operaciones. También resulta rilido, y a veces litil, volwr a dec:luu b mim atn'batae y aperaciones. Podemoe hacerlo para aignar un atributo de la subclue, un valor diferente para el mismo atributo de su superdase o aplicar a una operación de la sabc;lue funcionalidad diferente con respecto a la misma openci6n de su superclue. Este reamo se denomina reemplazo.

Por ejemplo, si tenemos una clase A:

```
elaaa A
{
   var Sattribute • •clafault valua•;
   fmiction aperation(I
   {
      eclo •io.athi11<1cbr />'1
      echo •Tha •alua of \SattTibute ia Sthia->littributacbr />";
```

y queremos alterar el valor predeterminado de $attribut• t~trar nueva funcionalidad para operation (), podemos crear la siguiente !Le 8, que reemplaza $attribute y operation ():

```
el••• a axtandll A
   var $attribute • 'diffarent valua';
   functioa operationl1
   {
      Hho-'S-thing-el"cbc   />' ;
      echo •Tha valua of \Sattribute ia Stlaia->&ttributacbr />"1
```

La declaración de B no afecta a la defuudón original de A. Considere lu siguientes dos líneas de código:

```
Sa • -   ACI;
Sa -> oparationl1,
```

Hemos creado un objeto de tipo A y hemos llamado a la función operation ( ). El resultado sen:

```
S-thing
Tha valua of $attribute ia dafault valua
```

si la aeación de 8 no ha alterado a A. Si cnmnos un objeto de tipo 8, obtendremos an resultado diferente.

El siguiente código:

```
Sb ·-•11,
Sb -> oparati-11;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

generan

```
s —l:lu.119
'l'he value of sal:l:ribul:e i• differeut: value
```

De la misma forma que el suministro de nuevos ab'ibutos *y* operac:ioneadentro de una subclue no afecta a la superdase, el reemplazo de atributos u operaciones dentro de una subclase no afecta a la superdase.

Una subclase heredará todos los atributos *y* operaciones de su superdue, a menos que se suministren reemplazos. Si suministra una definic:ión de reemplazo, &ta tend,rá preferencia *y* reemplazará a la definición original.

La palabra clave parent le permite invocar la versión original de la operación en la clase principal. Por ejemplo, para invocar A: :operation desde la clase B, tendría que utilizar parent: :operation () ; • Sin embargo, el resultado generado es diferente. Aunque invocamos la operación desde la clase prindpal, PHP utiliza los valores de ab'ibuto de la clase actual. Por ello, se obtiene el siguiente resultado:

```
8 —l:lu.119
'l'he value of Sal:l:ribul:e ia diffaraut: value
```

La herencia puede tener varias capas de profundidad. Por ejemplo, podemos declarar una clase e, que amplie B *y* que por tanto herede las funciones desde B *y* de la clase superior A. La clase *e* puede, a su vez, selecdonar ab'ibutos *y* operaciones d,esus clases principales para anularlas *y* reemplazarlas.

Evitar la herencia y los reemplazos con final

Una del~ no~e.<J:a~esde —S es lapa~bra cla\'e Unal . Al utilizarla por delante de la dedaraci6n de una función, &ta no se podrá reemplazar en ninguna clase. Por ejemplo, puede añadirla a la clase A del ejemplo anterior:

```
claaa A
(
  -r Sal:l:ribul:a • 'default: valua•;
  u-l fuucl:ioa operal:l- O
  (
    acbo 'S —l:hi1lgcbr />•;
    ecbo •'!lle value of .\Sal:l:ribul:e ia Sl:hia·>at:l:ribut:ecbr />";
```

Al aplicar este enfoque se evita que se reemplace operation() en la clase B. Si intenta hacerlo, obtendrá el siguiente error:

```
••cal erzor, ca—l: override u—l —t:llod A: ,aperacioaCI
```

Tambim pued~ utilizar la palabra clave final para evitar que se creen subclases de una clase. Para evitar que se generen subclases de la clase A, puede hacer lo siguiente:


```
f1-l claaa A
(...)
```

Si tras ello intenta heredar de A, un obtendrá un error:

```
••hl —— · Clua 8 •Y DOI: iaherit fra. f1-l claH (AI
```

Herencia múltiple

Algunos lenguajes orientados a objetos (en especial C++ *y* Smalltalk) admiten la herencia múltiple pero no u(PHP. Esto significa que cada clase sólo puede heredarse desde una clase superior. No existen restricdones en cuanto a la cantidad de clases secundarias que puede compartir una clase principal. Puede que este concepto no resulte claro al principio. La figura 6.1 muestra tres formas diferentes de derivar las clases A, B *y* C.

..... llmple.

Figura 1.1. PHP no admite la herencia nwltlple.

La combinación de la izquimla muestra la clase e que se hereda de la clase B, que a su vez se hereda de la clase A. Cada clase tiene un elemento superior al menos, por lo que se trata de una herencia simple perfectamente factible en PHP.

En la combinación central, la clase B *y* la clase e se heredan de la clase A. Cada clase tiene una clase superior como máximo, por lo que de nuevo estamos ante una herencia permitida en PHP.

La combinadón de la derecha muestra la clase e que se deriva de la clase A *y* de la clase B. En este caso, la clase e consta de dos clases prindpales lo que la convierte en una clase no válida en PHP.

Implementar interfaces

PHP S presenta el concepto de interfaz. Se trata de soludones para la herencia múltiple *y* son similares a la implementaciónde interfaces de otros lenguajes orientados a objetos, como Java.

EL_HOLISTICO@HOTMAIL.COM - 2006

Una interfaz específica una serie de funciones que deben implementane en las clases que implementen dicha interfaz. Por ejemplo, puede que desee una serie de clases que se puedan mostrar a sf mismas. En lugar de disponerde una clase principal c~ una función display(l de las que todas se hereden y se reemplacen, puede amplementaruna función de esta forma:

```
iacerface  Di•pl•yaole
(
   fuaccioa di•Pl•YII;

class webPage implements Displayable
{
   fuaccioa  di•Pl•YII
   (
   II
   }
```

Este ejemplo ilustra un tipo de herencia múltiple ya que la clase webPage se puede heredar de una clase e implementar una o varias interfaces.

Si no implementa los métodos especificados en la interfaz {en este caso display< I), se generará un error muy grave.

Diseñar clases

Ahora que ya conocemos algunos de los conceptos en los que se basan los objetos y lu ciase,,-y la sintaxis para implementarlosen PHP, ha llegado el momento de examinar c:ómo diseilar clases útiles.

Muchas clases de nuestro código representan clases o categorías de objetos del mundo real. Las ~lases que se utilizan en el desarrollo Web pueden incluir páginas, componentes de interfaces de usuario, carros de la compra, resolución de errores, categoriu de productos y clientes.

Los objetos de su código tambi6n pueden representar instancias específicas de clases mencionadas pre~mente,por ejemplo, la página principal, un botón dado o el carro de la compra utilizado por Fred Smith en un momento dado. El propio Fred Smith puede representane mediante un objeto de tipo cliente. Cada artfc:ulo comprado por Fred se puede representarcomo un objeto que pertenezca a una categoría ociase.

. En el capitulo anterior, util~mos sencillos archivos de inclusión para propor-a~r a nuestra c~mpai\fa fi~a, TLA Consulting, un aspecto visual y operativo uniforme en las diferentes páginas del sitio Web. El uso de las clases y la herencia pe~té crear ve~iones más avanzadas del mismo sitio.

Queremos disponer de la posibilidad de crear páginas rápidamente pan el sitio TI.A que se comporten de igual forma y presenten un aspecto semejante. Estas páginas se podrút modificar para su adaptacióna diferentespartes del sitio.

Vamos a crear una clase Page. El objetivo global de esta clase es limitar la cantidad de código HTML necesario para crear una nueva página. Nos permitirá alterar las partes que cambian de una página a otra además de generar automáticamente los elementos para que no varíen. Las clases ofrecen un marco flexible para erar nuevas páginas y no deberían comprometer nuestra libertad de acción. Como estamos generando la página desde una secuencia de comandos en lugar de código HTML estático, podemos agregar cualquier elemento que resulte inteligente, incluyendo funcionalidad para:

- Permitimos alterar elementos de página en un lugar. Si cambiamos la advertencia sobre los derechos de autor o agregamos un botón adicional, sólo necesitaremos realizar el cambio en un punto.
- Disponer de contenido predeterminadopara la mayor parte de las páginas, pero disponer de la capacidad de modificar cada elemento donde resulte necesarioy establecer valores personalizados paraelementos como el titulo y las etiquetu meta.
- Reconocer qué página se est6 visualizando y alterar los elementos de navegación para adaptarlos {no tiene mucho sentido tener un botón '!inculado a la página principal dentro de la página principal).
- Permitimos sustituir elementosestándar para páginas particulses. Por ejemplo, si queremos utilizar diferentes botones de navegación en las secciones del siti.o. dtbe~PJ 4~s_P9~~e ~ ~i~i~ad ~es~~~r _l~~tá~~ar.

Escribir el.código para nu.es.tra__clase_

Tras decidir el aspecto que queremos que devuelva nuestro código y la aplicación de una serie de funciones, ¿cómo lo implementamos?En una parte posterior del libro abordaremos el diseño y la gestión de proyectos de gran tamaño: Por el momento, vamos a concentramos en las partes espedficanelacionadas con la escritura de PHP orientado a objetos.

Nuestra clase necesitará un nombre lógico. Como representa a una página, se denominará Page. Para declarar una clase llamada Page, escribimos

```
cla•• P•CJ•
(
}
```

Esta clase necesita algunos atributos. Estableceremos los elementos que es probable que queramos modificar de una página a otra, como los atributos de nuestra clase. El contenido principal de la p6gina, que será una combinación de etiquetas HMTL y texto, se denominará $content. Podemos declarar el contenido con la siguiente linea de código dentro de la definiciónde clase.

```
public  Scoaceac;
```

Tambiél podemos establecer los atributos para almacenar el título de la página. Es probable que queramos modificar este elemento para indicar con claridad la página que está visualizando nuestro visitante. En lugar de incluir títulos en blanco, suministraremos un título predeterminado con la siguiente declaración:

```
public Stitle • "rLA coneulting Pty Ltd';
```

La mayor parte de las páginas Web comerciales incluyen etiquetas meta para ayudar a los motores de búsqueda a indexarlas. Para que resulten útiles, las etiquetu meta deberían cambiar de una página a otra. De nuevo, suministramos un valor predeterminado:

```
public $keyworde • 'TI.A Coneulting, Tllree Letter Abbreviation,
      8CIIIe of ay beet friend• are •••rch enginee•;
```

Los botones de navegación que se muestran en la página original de la figura 5.2 (vúse el capítulo anterior) deberían mantenerse entre lu diferentes páginu para evitar confundir a los usuarios, pero para poder cambiarlos con facilidad los convertiremos en un atributo. Como el número de botones puede ser variable, utilizaremos una matriz y almacenaremos el texto del botón y del URL al que debería apuntar.

```
public Sbuttona • array( 'Home•      •>  'bome.php',
                         •contect'      •>  •contact.pbp•,
                         •servicee  ->  'Mrvic-.pllp,
                         'Site Map'  •>  'map.php'
                        );
```

Para proporcionar alguna funcionalidad, la clase tambit!n necesita operaciones. Podemos comenzar porsuministrar funciones de descriptores de acceso para establecer y obtener valores de los atributos definidos. &tos suelen adoptar la siguiente forma:

```
public function ~ ..tl$name, $value)
 (
   $thie->$name • $value;
```

La función _set () no contiene comprobación de errores (por motivos de brevedad) pero la podremos añadir más adelante. Como es poco probable que solicitemos alguno de estos valores desde fuera de la clase, puede optar por no utilizar una función _get ().

La función principal de esta clase es mostrar una página de código HTML, por lo que necesitaremos una función. Hemos llamado a la función Display () y presenta de la siguiente~:

```
public fUDCtioa Dieplay()
   (
      ecbo •chtlll>\Bcllaad>\n•;
      Sthie -> DieplayTitlel);
```



```
      $tbi• ·> DieplayJCeyvord8(11
      $th1• ·> DieplayStyleell1
      ecbo •c/head>\ncbody>\n•;
      $th1• ·> D1eplayHead8rl);
      Sthie -> DieplayNenul$thie->buttonal,
      ecbo Sthie->content;
      Sthi• ·> Dieplayrooterll;
      ecbo •c/body>\nc/htal>\n•;
```

Esta función incluye varias instrucciones echo sencillas para mostrar c6digo HTML, pero se compone fundamentalmente de llamadas a otru funciones de la clase. Como es probable que haya supuesto por los nombres, dichas funciones muestran partes de la página.

No es obligatorio dividir las funciones de esta forma. Todas ellas podrían combinarse en una función de mayor tamaño. En nuestro caso, las hemos dividido por varias razones.

Cada función deberla realizar una tarea definida. Cuanto más sencilla sea, más fácil resultará escribirla y probarla. Tampoco conviene excederse: si divide su programa en unidades demuiado pequeñu puede resultar dificil de leer.

El uso de la herencia permite reemplazar operaciones; Podemos reemplazar la fundón Display () , pero es poco probable que queramos cambiar la forma en la que se muestra toda la página. Resultará mucho mejor dividir la funcionalidad de visualización en unas pocu tareas independientes y disponer de la posibilidad de reemplazar .sólolas partes que deseemos modificar.

La función Display() llama a DisplayTitle (), Displaylteyworcls (), DisplayStyles (), DiaplayHeader (), DisplayMeDu () y DisplayFooter (). Por lo tanto, necesitamos definir estas operaciones. Podemos escribir operadones o funciones en otden"lógico. Para ello; se llama a la operacióll'o-. la función antes que-- al código de la fundón. En muchos otros lenguajes es necesario escribir la fundón u operación antes de poder llamarla. La mayor parte de nuestras operaciones son butante simples y necesitan mostrar HTML y quizú los.contenidos de nuestros atributos.

El listado 6.1 muestra la clase completa, que hemos guardado como pag~. inc para incluir o exigir dentro de otros archivos.

Llatado 6.1. pege.lnc. Ladese Page proparc:lal• uns fonN ftc:11 y flexible de cnar s-glnu psrael sitio TLA.

```
c?php
el••• Page
(
   // atributoe de lael•••  Pag-e
   public $content;
   public $t1tle • 'TLA CODeultizlg Pty Ltd';
   public $keyword• • •TLA con-lting, Three Letter Abhreviatioa,
        -    of ay beet friend• are.eearch eaginee•;
   public $button• • array( ·-·        •> 'hoae.pbp•,
                            •CODtact•   •> •contac:t.php•,
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
                    '8eni-·        ...·NJ:Ti-.....
                    •11ee ...,.    •..  ·-p.ptip•
        ) /

. // openoi-        de la clue ....
  public f,aacti•     _Mt    1$-,     S•aluel
  (
     Stllie· .....     •  .. al__,


  plllaUc  faactiGa'Di9play(I
  (
     ectlo  •clltlll•\Dcllead  ..\a•,
     Stllia-  .. Di11Plaw'l'itle(Il
     ftllia  • ., DillPlQlll1Wda()  1 ·
     ·~      -,  Di ..  k&J9t:)'I–U/
     ectlo  •c/.....,\IICIIDdp\a•,
     tcllla   - .. Diapl.Q9Mder O'
     niila ~.. m.,ia,...l l tü•·,...tt0a1,
     ec11o  .tJaia••  H--1
     Stllla-..   Di~(l1
     ..._   •c/lNldf'>\–,-.ia.\11•1


  pdllc  l!lulctlÍIIIIJtllplayftt"la(l
  (
     ecllo  •ctitle  ..  '.Stlü.a-  ..tltle.  •  e/U.u.  ..•,
  ¡  ....

  pulilic  hllcti•   ~(I
  '  r-- ·-ta-  ·  ~'  ~-e-
  ...-·-·-  -----~:tUaUUl:Jai"F ~~~?-7.,•¡
  )

  palt1ic   f1mcti•   Dillplay9CylH()
  (
1..
  cat)'le ..
el•·
  b1 (color,wllite,    faot-aise:24ptl    tmmt·allga:ceater,
     f– - f..Uyrarial, — ·HrU}
  .-      (colori14aite,   fCl1lt•ai  ..  ,1apc,   t..C•aliga,caater,
     f–t·f-.d.ly|arial,  .....  ·aar1fl   faat-weigllt,bold)
  td  (~sbladi:)
  p  (-lordtlac:Ju   t– - •ise|Up&1   tAa·alig11::l11aCUy,
     f~·faaiilysar~al,  .....  ·Mrif)
  p.f-      (color,wllite,   faat-ai-1tpt1       t..C•aligau:~ar,
     f–t·fad.lysuial,  ..... U /  faat-–igllt:bold)
  a,l1–,a,naitad,a,acti      ..   (eolor|wllite)
--,
  cfffylal>
e,....
  )

  public function DisplayHeader()
```



```
1..
  ct:altle  ——  - 100l-•  -11~          •"12"  .u.,.ll ....  ,.  ilosMr  ••, ...
  cu   11gao1ar  .·111ut• ..
     ctd  aliga   •"laft"»ciml   are-.  "lClfD,vir    / ..c/td.,
     ctd.,
        cllb~      CCIUl&ltial}   Ltdc/bb
     c/ttb
     cll:daliga   •"rlglat•..cim9   .re  •  •loglt.gU•  /..c/td»
  c/tn
  c/Hbla ..
c?pllp
  }

  pabUc   fuactlaa    Diapla,...__,,..ttmal
  (
     ec11D  •ctllble   wldcJl,a·~·   ———·lllllit:e'   –u.ad# ...  '4'
            c:e.119plcia9-···..,••,
     ecllo  •  ctn\a•;

     llcalnJ.•  el   t:aM8o del  llotem
     .widtJI  •  1H1–Cl__..t___l  t

     r-=        llbatccma -   •-·     ...arll
     {
        Stlli8   •., DiaplayMll-(SWidtll~      8–,.  8-Vl,
                  lSt111a- ..lauar.c:mnath9e<Sur1J  ll
     }
     echo " </tr>\n";
     acllo  •cft:able  ..\a•,
  }

--,..u:cr.taal:tto1rn~          ...   ftllñ)-- -·----------------
  (
     iUatzpoaCS_1..,,..l',..__1SLP•J,      Sad    1-ralaal
     (
        retan     falael
     }
     alae
     {
        retara     trua,


  ,-u.e   fuiu:daa   Diaplay8utt:aaltwidtll,    •—·   S-1,  Saau ..   •  tnel
  (
     U   CSactl ..  1
     (
        acllo  •ctd   widl:la  •' • .lltal91titiMC8widtlll.   •1•"
           ca  llzef   •'",lltalaatiti   ..  (Sllrl).••»
           cim9  .re  •'•·logo.pf'      ali:.  •'"  .lltal—iti   ..  (——t.••  1111S1Nr
           •'D'    />e/a..
           ca  llref   .,~.lltalatitlHCSarll   ·'"..capaa  él ...  ·•-·      ...~/apm>
           cl–ltd.."l
     }
     .1 ..
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
        ecbo •ctd widtb ·••.btaleatities(Swidtbl.o1°>
           ci.. •re •'•ide·logo.9if'>
           capan cl••••'..DU'>Sn&mec/span>c/td>";


   public functioa DisplayPooter|I
   (
?>

      ctabl• widtb • •1001• bgcolor •"black• cellpaddi.og •"12° borde~ •"0">
      cCr>
         ctd>
            cp cla••••foot°>6copy, TI.A c-inaltia9 fty Led.c/p>
            cp cla••··foot°>Pl••••••• our
                 ca bref •"legal.pbpo>lec¡al iaforwiatiOD. pagec/&>c/p>
         c/td>
      c/tr>
      e/table>
c?pbp
   |

?>
```

Al leer esta et.se, ffjese en que DisplayStyles (), DisplayReader () y DispJ,ayFooter O necesitan mostrar un gran bloque de HTML estitico sin ningún procesamiento de PHP. Por lo tanto, sólo hemos utilizado una etiqueta final de PHP (?>), hemos escrito nuestro código de H1ML y hemos vuelto a introducir PHP con una etiqueta de apertura de PHP (e?php) dentro de las funciones.

En esta clase se definen otras dos operaciones. En concreto, la operación Displ-ayButton+>muestra unbot6ndemenú.Sielbot6hv|fU.puntara1a-¡,ágina en la que estamos, mostramos un botón inactivo que presenta un aspecto ligeramente diferente sin vinculación. De esta forma, logramos mantener la uniformidad de la página y suministramos a los visitantes una ubicación visual.

La operación IsURLCurrentPage () determina si el URL asociado a un botón apunta la p6gina actual. Se pueden utilizar muchas Ncnicas para descubrirlo. En este caso, hemos utilizado la función de cadena strpos () para determinar si el URL dado está incluido en una de las variables establecidas de servidor. La instrucción strpos($_SBRVER[PRP_SBLF'] ,$url) devolverá un número si la cadena de $url se encuentra dentro de la variable superglobal $_SBRVBR¢•PRP_SBLF•J o false en caso contrario.

Para utilizar esta clase de p6gina necesitamos ind uirpage•inc en una secuencia de comandos y llamar a Display ()•

El código del listado 6.2 creará la página de inicio de TLA Consulting y devolved un resultado muy similar al que generamos previamente en la figura 5.2. El código del listado 6.2 realiza lu siguientes tueu:

t. Utiliza ~equire para incluir contenidos de page•iDc, que contiene la definición de la clase Page.

2. Crea una instancia de la clase Page, denominada $hoeepage.
3. Define el contenido, formado por texto dinúnic:o y etiquetas HTML que aparecedn en la página (de esta forma se invoca implícitamente el método __set()).
4. Llama a la operación Display () dentro del objeto $bomepage para oblig11r a la página a mostrarse en el navegador del visitante.

Ustado 1.2. home.php. Estll i>'glns de inicio utlll:m Is daH P.ge psra realtzar la mayor parte del trabajo implicado en 18 generacl6n de 18 p6glna.

```
c?pbp
   requir• | 'P•9•.iac' | |

   Sllomepag• • -    ...   () /

   Sbollepage ·>coateac(•cp>..1C0lle to tbe bolle of TLA CoasultiDg.
           Pl•••• cake •ome ti.. to get to kaow u•.c/p>
           cp>We •pecializ• ia ••rviag you.r bu•iae•• aeed8
           aad bope to bear fr0II you oooa.c/p>'

   Sbomepage -> Di•play||¡
?>
```

Como puede observar en este listado, se necesita realizar muy poco trabajo para generar nuevas páginas utilizando la clase Page. El uso de esta clase de la forma ilustrada implica que todas nuestras páginas deben ser muy similues.

Si queremos que una sección del sitio utilice una variante de la página estándar, ~ta,._cg11~W~R&9•...in~tm lll\n\leYO_archiv.o.llamadopage2~ inc y. realizar algunos cambios. Como resultado, cada vez que actualicemos o modifiquemos partes de la página page•i.nc necesitaremos acorcla!JIOS de realizar los mismos cambios en page2•inc.

Una opción mejor consiste en utilizar la herencia para crear una nueva clase que derive gran parte de su funcionalidad de Page y que reemplace las partes que necesiten ser düerentes. En el sitio de TLA, queremos que la página de servicios incluya una segunda barra de navegación. La secuencia de comandos que se muestra en el listado 6.3 realiza esta tarea mediante la creación de una nueva clase llamada Serv:i,ceaPage que se deriva de Page. Suministramos UM nueva matriz llamada $row2l)uttons que contiene los botones y vínculos que queremos en la segunda fila. Como queremos que esta clase se comporte pdcticamentie de la misma manera, sólo anulamos la parte que queremos modificar: la operacida Display().

Usmdo u. HMCeS.php. u ptgill8 de servicios .. deriva de .. clase Pega pen, l'ffffl pl8n Qtaptey() para vartar et l'NUltado.

```
c?pbp
   require (•page.iac'1l
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
el••• Service•Page extend8 Paga
{
   private Sraw2button• • array( 'Re-engineering• •• •reengineering.php',
                 'Standard• Cotlpliance• •• '•tandarda.php',
                 'Buzzward Campliance• •• 'buzzward.php',
                 'Mi••ian Stat...nta• •• 'mi•sian.php'
            );
   public functian Di•play(I
   (
      echa •cht•l•\nchead>\n•;
      Sthi• ·> Di•playTitle1l;
      Sthi8 ·> Di•playXeyward8();
      $th1• ·> Di•playStyle•II;
      echa •c/head>\ncbady>\n•
      Sthi• ·> Di•playHeaderll;
      Sthi• ·> Di•playMenulSthi• ··buttan•I;
      Sthi• ·> Di•playMenul$th1• ··rav2buttan•I;
      eclia Sthi• ··canter.t;
      Sthia -> Di•playFaoterll;
      echa •c/bady>\nc/html>\n•;


S•ervice• • new Service•Paga11;
Saervicea ·• cantent •'<P>At TLA Calll1ulting, wa affar a nUllbar of 8e:vice•.
             Perhap• tha praductivity of your a.play••• wauld
            iaprove if wa ra-anginaared your bu•in••• ·
            Maybe all yaur busin••• needa 1• a fra•h •i••ian
            atate..nt, ora naw batch of buzzward8.c/p>';
S••rvic•• ·> Di•playll;
```

El reemplazo de Display () es muy similar con la excepción de que contiene una línea más

```
Sthi• ·> Di•playMenul$thi• ··raw2buttons1;
```

p~rallamar a DisplayMenu() una segunda vez y crear una segunda barra de menu.

Fuera de la definición de clase, creamos una instancia de la clase ServicePage, establecemos los valores para los que no queremos los predeterminados v llamamos a Display (). Como se ilustra en la figura 6.2, tenemos una nueva variante de la página estándar. El único código nuevo que necesitamos escribir es el que hace referencia a las partes diferentes.

La creación de páginas a travn de clases PHP presenta ventajas claras. Con una dase que realice la mayor parte del trabajo por nosotros, necesitaremos realizar menos trabajo Pª!8 crear una nueva pá——. Podemos actualizar todas las páginas de una.vez actuahzando la clase. La herencia nos permite derivar diferentes versíones de la dase desde el original sin perder sus ventajas.

Como~ con casi todo en esta vida, estas ventajas tienen un coste. La creación de páganas desde una secuencia de comandos exige un mayor esfuerzo del

procesador que la simple carga de una página HTML estándar desde el disco y su envio a un navegador. En un sitio con tráfico este hecho resultará importante por lo que convendría utilizar páginas tITML estáticas o almacenar en cache! el resultado de las secuencias de comandos allí donde resulte posible para cargarlo en el RrVidor.

Figura 6.2. La página de sefVic:ios se creó ullllzando la herenc:ia para volier a ullllzar gran parta da nuestrapáginaeslindar.

Nuevas funciones avanzadas orientadas a objetos de PHP 5

En los siguientes apartados analizaremos las funciones orientadas a objetos avanzadas de PHP, muchas de las cuales son nuevas en PHP S.

PHP 4 frente a PHP 5

Si ha utilizado anteriores versiones de PHP debe tener en cuenta algunas diferencias importantes.

En PHP 4, los objetos se pasaban por valor y ahora se pasan por referencia. Esta forma de escribir el código no deberla afectar a su código anterior y, de hecho, muchos programadoreshan escrito código ineficaz sin saberlo. Por ejemplo, incluso al escribir

```
$c: • new ayCla••1
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Se crea una nueva clase y se copia la instancia a $e (con lo que se crean dos copias aunque inmediatamentese pierde el puntero a una de ellas). Este comportamiento puede provocar problemas si asume que los objetos se pasan por referencia, en especial cuando se pasan a funciones. La mayorfa de los lenguajes orientados a objetos pasa los objetos como referencias de forma predeterminada, entre los que se incluye PHP. La otra gran diferencia es que anteriormenteen PHP era complicado anular las referencias a objetos devueltos por funciones, sobre todo para invocu mitodos en dichos objetos. Antes no se podfa escribir algo como

```
••lec:t_obj~Cl->di11playCI;
```

donde select_obj ect (› devolvfa un objeto con un atodo display (). Esta instrucción funciona ahora sin problemas.

Utilizar constantes de clase

PHP presenta el concepto de constante de clase, que se puede utilizar sin necesidad de crea una instancia de una clase, como se muestra en el siguiente ejemplo:

```
c?pllp
claH  Nath {
   conllt pi•  3.14159;.
;)
ecbo  'Nllth, :pi •  '.Nllth: ,pl.  •\n• 1
h.
```

Puede acceder a la constante si utiliza el operador : : para especificar la clase! a _l!_gy_e~ la consante. Joroo.hemos.hecho en el ejemplo_

implementar métodos estáticos

Otra de las novedades de PHP es la palabra clave static. Se aplica a los ato-' dos para que puedan invocarse sin creu una instancia de la clase. Es el equivalente para mitodos del concepto de constante de clase. Por ejemplo, tomemos la clase Math creada en un apartado anterior. Podrfa añadir una función squared () a la misma e invocarla sin crear una instancia de la clase:

```
el••• Nath
{
   •tatic functiOD mquaredl$input1
   {
      ratuas $input•$input1

}
ec:Jlo Nllth::mquared(lll
```

Sepa que'no puede utilizar la palabra clave thia dentro de un mftodo estjtico ya que no puede haber una instancia de objeto a la que hacer referencia.

Comprobar el tipo de clase y sugerir tipos

En PHP 5 aparece por primera vez la palabra clave inatanceof y el concepto de tipos de clase.

La palabra clave instanceof nos permite comprobar el tipo de un obj~. Podemos comprobar si se trata de una instancia de una determinada clase, u ~ hereda de una clase o si implementa una interfaz. Esta palabra clave es un eficaz operador condicional. Por ejemplo, con los ejemplos anteriores en los que implementjbamos la clase B como subclase de A. obtendrfamos:

($b instanceof Bl seria true.

($b instanceof A) seria true.

($b instanceof Displayablel seria false.

Todos estos ejemplos asumen que A, B y Displayable se encuentran en el 6mbito actual; en caso contrario se generarla un error.

Otra de las novedades de PHP 5 es la sugerencia de tipos de clase. Por lo general, al pasar un parllmetro a una función, no se pasa el tipo de dicho pullmetro. Con la sugerencia de tipos de clase, puede especificar el tipo de clase que d~berfa ~sarse y, si no es el que realmente se pasa, se genera un error. La comprobaoón de tipos es equivalente a instanceof. Veamos la siguiente función:

```
fuaction cbacll:_bint11 $—1 • • • 1
{
   // ...
```

Este e¡éinj,lósiigiire que $aoineclili• tiene que ser una instincia d~la c1ae

Si pasamos una instancia de la clase A como

```
chec:ll:_hintC$al;
```

se genera el siguiente error:

```
ratal error, ~nt      1 ...  t be an in•tanc:e of 8
```

Si hubiera sugerido A y pasado una instancia de B, no se habrfa prodl,!,cid~ un error ya que B se hereda de A.

Clonar objetos

PHP 5 presenta la palabra clave clone por primera vez, pua poder copiar un objeto exiatente. Por ejemplo,

```
$c = clone $b;
```

aN una copia del objeto $b de la misma c:lae, con los mismos valores de-•tributo.

EL_HOLISTICO@HOTMAIL.COM - 2006

•Tambim puede CU1biar este comportamiento. Si necesita mi com~ento no predeterminado para clone, tendrá que c:_rear un método en la clase ——con4_I nombre _ clone (), método que es similar a un constructor o un destrue!O~r-Y~!; que no se invoca directamente. Se invoca cuando se utiliza la palabra clav4i ~lC?~'" como indicamos en este apartado. Tras ello, dentro del m~ ;_cloae(l.——e definir exactamente el tipo de comportamiento de cop.ia que desee.

Lo mejor de _ clone (›J:5 que se invocad tras creane una copia exacta por medio del comportamiento predetemünado, por lo que sólo podrai cambiar lC? ciuie desee cambiar.

La funcionalidad más habitual que se añade a_ clqne (› es c6digo para garantizar que los atributos de la clase que se procesan com~ relerenc:;as se copian eerrectamente. Si desea clonar una clase que contiene una referencia a un objeto, probablemente espere una segunda copia de dicho objeto en lugar de una segunda referencia al mismo, por lo que tendria sentido añadirlo a _clone (›.

Tambim puede optar por no cambiar nada y realizar otra acción diferente, como por ejemplo actualizar un registro de base de datos subyacente relacionado con la clase.

Utilizar clases abstractas

Otra de las novedades de PHP 5 son las clases abstractas. Estas claies no se puiNen ms tan aar.

PHP 5 tambim o&ece métodos abstractos, que proporcionan la firma de un método pero no su implementación, como se muestra en este ejemplo:

```
uacr~~t operationXCSpar-1, Spar-21:
```

Cualquier clase que contenga métodos abstractos debe ser abstracta, como se indica a continuación:

```
ab•tract el••• A
1
abatract fwaction operationXISparUJ., $paraa21:
```

i~ii_~dases y métodos abstractos se suelen utilizar en jerarquías de clases complejas. Si quetodas las subclases d~ incluir y reemplazar un mflod.o concreto, lo qu~lffl~ ~ puede hacer con una interfaz.

s~¸~~¿argar métodos con =-call()

Anted.orpiente hemos visto diferente, métodos con signüicados especiales cuym n~ empezaban por un doble guión bajo LJ, como por ejemplo__get (›, _•et(), _corustruct () y __de•truct (). Otro e;emploes el método __call (), que se utiliza en PHP para implementar la sobrecarga de métodos. La sobrecarga de métodos es habitual es muchos lenguajes orientados a objetos pero en PHP no

resulta demasiado iltil ya qúe en su lugar se suelen utilizar tipos flexibles y los parámetros opcionales de las funciones.

Para utilizarla, debe implementar un método_ call (), como se indica a continuación:

```
public fwaction _ calll$..tbod, ~pl
{
   if 1$..thod •• 'di•play
      if li• ob:l•ct1$p(0J ll
         $thi•›di•playOhject1$p(Oll
      •l•• if li• •rrayl$p(Oíll
         $thi•›»dī•playArray($p(Oll;
      •l••
         $thi•›di•playScalar($p(O)I
```

El método_ call () adopta d0l5 paraimetros. El primerÓ contiene el nombre del método invocado y el segundo una matriz de los parúnetros pasadm a dicho método. Puede decidir qué método subyacente invocar. En esté caso, si un objeto se pasa al método display (), se invoca el método sub~te displayObject (); si se pasa una matriz, se invoca displayArray (› fsi se pasa otro elemento, se invoca di splayScalar (). Para invocar este código, primero debe crear una instancia-de la clase que contenga este método __call (› (por ejemplo con el——overload) y, tras ello, invocar el método display(), como se indica a ~tinuación:

```
$ÓV • new overload
Sov-»dl•play!array(1, 2, JII 1
$OV›dl›play( 'cat' I:
```

La primera llamada a display () invoca displayArray ().r_la~ in-va-ca displayScalar () . No necesita una implementación subyacente del método display () para que este código fundane.

Utilizar _ autoload()

Ptra de las funciones especiales es_ autoload (›• No se trata de un método de clase sino de una función independiente, es decir, se declara fuera de cualquier declaración de clase. Si la implementa, se invocar, automáticamente al intentar crear una instancia de una clase que no se ha declarado.

El principal uso de autoload () consiste en intentar incluir cualquier archivo necesario para crear una instancia de la clase requerida. Veamm •te ——lo:

```
function __autoload($name)
{
   include_once $name.'.php';
}
```

Esta implementación intenta incluir un archivo con el mismo nombre que la clase.

EL_HOLISTICO@HOTMAIL.COM - 2006

Implementar iteradores e iteración

Una de lu características del nuevo motor orientado a objetos es que podemos utilizar un bucle foreach() pan iterar por los atributos de un objeto como si se tratara de una matriz. Veamos un ejemplo:

```
el••• myCl•••
(
   publie Sa.S1
   publie Sb•71
   pullle  Sc-t,

Sx • aew ayCl•••;
Coreac:b !S& aa Sattriblltel
   ecbo Sattribv.te.•cbr /»•;
```

Al cierre de esta edición, el manual de PHP sugerfa la necesidad de implementar la interfaz vada Transversable para que funcione la interfaz foreacb,, pero si lo hacemos se genera un error muy grave. El funcionamiento parece correcto si no se implementa. Si necesita un comportamiento mú sofisticado, puede implementar un iterador. Para ello, hacemos que la clase sobre la que qaeramoe iterar implemente la interfaz IteratorAggregate y le asignamos el mftodo getIterator, que devuelve una instancia de la clase de iteración. Esta clase debe implementar la interfaz Iterator, que cuenta con una serie de mftodm que es necesario implementar. En el listado 6.4 se recoge un ejemplo de una clase y un iterador.

Uatado 1.4.. lterator.php. Una sencilla dase bue y una dale de lteradón.

```
--cJpbp
el••• Objeetiterator imple..nt• Iterator (

   private Sobj,
   private Seount;
   private sc:urrentilldex1

   fUIIClIiOD _eonatruetlSobj)
   (
      Stbi•-»obj  • Sobj;
      Stbia »eoasat • eount(Stbia-»obj-»data);
   }
   f'UDCtion rewilldl>
   (
      Stbia-»c:urrentilldex • 0;
   }
   i-uoa     validO
   (
      return $this->currentIndex < $this->count;
   }
   function key()
   {
      return $this->currentIndex;
   }
```



```
   function current()
   {
      return $this->obj->data[$this->currentIndex];
   }
   function next()
   {
      $this->currentIndex++;
   }

elaaa Object impl...nta IteratorAggregate
{
   pablie Sdata • array() 1

   f11DCtion _ eonatruetlSinl
   {
      Stbia-»data  • Sin;
   ) .

   f\111Cticm getiterator[l
   (
      return new Objeetiterator(Stbial;


s-,object.  uw Object(array(2, •  ,,  1,  1D11;

S11Yiterator • SmyObjeet-»getiterator(I;
for(S-.yiterator-»rewilld(l;  Smyiterator-»valid(l;  Smyiterator-»next()I
(
   Skey • Smyiterator »key();
   Svalue • Smyiterator-»eurrent() 1
.cIIo  $kev  •>  Svalue cbr r»•,
}
```

La clase Obj ectiterator cuenta con una serie de funciones que la interfaz Iterator necesita:

- m constructor no es necesario pero conviene utiliznlo pan definir valores para los distintos elementos sobre los que deseemos iterar asf como un enlace al elemento de datos actual.
- La función rewind () debe establecer el puntero de datos interno en 1'parte inicial de loe datos.
- La ñmdón valid () debe indicar si siguen existiendo datos en la ubic,ción actual del puntero de datos.
- La función key () debe devolver el valor del puntero de datos.
- La func:i6n value () debe devolver el valor abnacenado en el puntero d datos actual.
- La función next () debe desplazar el puntero de dala9i,or:loii datm.

El motivo de utilizar una clase de iteración como nta es que la interfaz a los datos no cambiaril aunque varíe la implementad6n subyacente. En este ejemplo, la clase IteratorAggregate es una sencilla matriz. Si decide cambiarla por una tabla de hash o una lista vinculada, podrfa utilizar un elemento Iterator estándar para recorrerla, aunque el c6digo de Iterator cambiaría.

Convertir clases en cadenas

Otra de las nuevas funciones m,gicas es __toString (l. Si la implementa en una clase, se invocar, al intentar imprimir la clase, como se indica en el siguiente ejemplo:

```
$p • n.., Printallle:
ecbo $p,
```

Lo que devuelva la funci6n __toString () se imprimiril por medio de echo. Puede implementarlo de esta forma:

```
el••• Priatable
{
   var $teatone,
   va:r $teattwo,
   public eunctioa ~tOString11
   '{
   i-  :retura(var_export(Sthia,  TRIJBII,
```

(La funci6n var_export O imprime los valores de todos los atributos de una clase.)

Utilizar el API de reflexión

Otra de las nuevas características de PHP 5 es el API de reflexi6n. La retlexi6n es la capacidad de inteaogar clases y objetos existentes para determinar su estructura y sus contenidos, lo que puede resultar muy útil cuando nos enfrentamos a clases desconocidas o sin documentar, por ejemplo a secuencias de comandos de PHP codificadas.

El API es realmente complejo pero analizaremos un ejemplo para que se haga una idea de cómo utilizarla. Retomemos la clase Page que definimos antes. Por medio del API puede obtener toda la infonnacidn sobre esta clase,. como se indica en el listado 6.5.

Listado 6.5. reflection.php. Mostrar información sobre la clase Page.

```
c7pllp
requi:re_oace('PA99.1Ac:•1,
$claee • aew ReflectioaCleaa(•Page•);
```



```
ec:bo • cpre>• /
echo Scleaa,
echo 'e/pre>':
?>
```

En este caso, utilizamos el método _ toString O de la dale Reflectioa para imprimir los datos. Las etiquetas cpre> se encuentran en lfneu diferentes para no confundir el método _toStringO: La primera pantalla del resultado generado por este c6digo se ilustra en la figura 6.3.

Flgu,. e.3. El NtSUltmo del API de retlexlón es sorprendenlemen~ delallado.

A continuación

En el siguiente capítulo describiremos las nuevas func:iones ~ control de ex~p-dones de PHP. Las excepciones constituyen un elegante m~ ~ soluao-nar errores de tiempo de ejecud6n.

EL_HOLISTICO@HOTMAIL.COM - 2006

7

Controlar excepciones

En este capítulo analizaremos el concepto de control de excepciones y cómo se implementa en PHP. Las excepciones son una de las novedades más importantes de PHP 5. Constituyen un mecanismo muy útil para solucionar errores de forma extensible, mantenida y orientada a objetos. Nos centraremos en los sigui~tes aspectos:

- Conceptos del control de excepciones
- Estructuras de control de excepciones: try ... throw ... catch
- La clase Exception
- Excepciones definidas por el usuario
- Excepciones en el ejemplo Bob's Auto Parts
- Excepciones y otros mecanismos de resolución de errores en PHP

Conceptos del control de excepciones

El concepto básico del control de excepciones es el código que se ejecuta en el interior del denominado bloque try, bloque que tiene el siguiente aspecto:

```
try
(
   //  aqul  "a  el  c:6di90 .
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Si sucede algo incorrecto dentro del bloque t ry, se genera una excepción. Algunas lenguajes, como Java, generan las excepciones autom4ticamente en determinados casos. En PHP, las excepciones deben generarse manualmente. Las excepciones se generan de esta forma:

```
throw new Exception('mensaje', código);
```

La palabra clave throv desencadena el mecanismo de control de excepciones. Es_una construcá6n del lenguaje mú que una función pero será necesario pasarle un valor. Espera reábir un objeto. En el caso ms senállo, se puede crear una instancia de la clase incorporada Bxcept ion, como hemos hecho en este ejemplo.

El constructor de esta clase adopta dos parámetros: un mellsaje y un c6digo que, respectivamente, representan un mensaje de error y un número de c6digo de error. Ambos parámetros son opcionales. Por último, por debajo del bloque try necesitaremos al iiienos un bloque catch, con el siguiente aspecto:

```
catcll (typebint axc:eptionl
{
   // controle la excepci6n
}
```

Puede haber mú de un bloque catch asoáado a un mismo bloque try. El uso de mú de uno tendrfa sentido si cada bloque catch espera capturar un tipo de excepción diferente. Por ejemplo, si desea capturar excepciones de la clase Bxception, el aspecto del bloque catch tendrfa el siguiente aspecto:

```
cat~ ISXC:..!J!tion ,e,

   /( controle la exc:epci6n
```

El objeto que se pasa al bloque catch (y que éste captura) es el que se pasa (y que genera) a la instrucción throw que provoca la excepción. La excepción pue~e ser de cualquier tipo, aunque es aconsejable utilizar instancias de la clase lxception o instancias de excepciones definidas por el usuario derivadas de esta clase. (M4s adelante veremos cómo definir excepciones personalizadas.)

Cuando se genera una excepción, PHP busca un bloque catch que coincida. Si tiene mú de un bloque catch, loa objetos pasados al mismo deben ser de tipos diferentes para que PHP pueda determinar quf bloque catch utilizar.

Otro aspecto que tener en cuenta es que se pueden generar m6s excepciones dentro de un bloque catch. Para que esta operación resulte más clara, analizaremos un ejemplo. En el listado 7.1 se reproduce un sencillo ejemplo de control de excepciones.

U.tado 7.1. balic_exceptlon.php. Generar y capturmr una excepción.

```
dpllp

try
```



```
   tllrow new llxcepeion('A terrible error baa occurred•, ul,
)
catc.b lkception Sel
{
   ec:bo 'llxception '• Se->getCOde(1• •: •• te•:,getNeeaage(1
      .•in'.   Se•:,getPile(I. ' oa li~  •. Se•"98tLiae(I.  •cbr j,,,,

,"
```

En este código hemos utilizado diferentes mftodos de la clase Bxception. que analizaremos en breve. El resultado de la ejecución de este listado se ilutra en la figura 7.1.

Figura 7.1. Este bloque catch informa del menIIIje de enar • indica dónd~ .. ha producido.

En el código, puede ver que generamos una excepción de la clase Bxception. Esta dase cuenta con métodos que podemos utilizar en el bloque catch para devolver un útil mensaje de error.

La clase Exception

PHP S incorpora UJ1' dase denominada Bxceptioa. Su c:onstruc:tor adopta dos parámetros, como hemos indicado antes: un mensaje de error y un c6dig~ de error.

EL_HOLISTICO@HOTMAIL.COM - 2006

Ademú del constructor, esta clase dispone de los sigui.mies ~os:

- getCode (›: Devuelve el c6digo tal y como se haya pasado al construdor
- getMeHage ‹›: Devuelve el mensaje tal y como se haya pasado al constructor.
- getFile () a Devuelve la ruta completa al archivo de c6digo en el que se ha producido la excepción
- getLine (): Devuelve el número de linea del archivo de código en que se haya producido la excpáón
- getTrace ( ): Devuelve una matriz con un rastro q:ie indica dónde se ha producido la excepción
- getTraceAsString (): Devuelve la misma infonnaci6n que getTrace, pero con formato de cadena
- _ toString (): Le permite reproducir un objeto Bxception y proporciona toda la información de los métodos anteriores

Apreciará que en el listado 7.1 hemos utilizado los cuatro primeros mftodos. Puede obtener la misma información (mú la pista) si ej~:

```
ecbO ••l
```

Esta pista 111UeStraqui funciones se ejecutan en el momento en que se produce la excepción.

Excepciones defi11idás porel usüario

En lugar de aear una instancia de la clase base Bxcept ion y de pasarla, podemos puar el objeto que deseemos. En la mayoría de los casos, ampliaremos la clase Bxception para aear nuestras propias clases de excepción.

Podemos pasar cualquier otro objeto a la cl6usula throw. Incluso podemos hacerlo si un determinado objeto nos da problemas y queremos pasarlo por motivos de depuración.

No obstante, por lo general ampliaremos la clase base Bxception. El 11\Ulual de PHP proporciona código que muestra la estructura de esta clase. Dicho código, obtenido de http://www.php.net/zend-engine-2.php, se reproduce en el listado *7:l.* No se trata del código real, pero representa lo que podemos esperar que se herede.

Listado 7.2. Clase Exception. Esto es lo que puede esperar que se herede.

```
dpllp
cl.Al9• Sa:epeioa (
```



```
fwac:t:l.oa __coastnct (etr:1.ng S••ave•IRILL, illt $code-Ol (
   if (func_mnl_arga(ll (
      $thia ·>••aage • S•••aga;
   |
   $thi8•>Code • $code;
   $th:l.a•>fil• •   FILS  ,   //dela clluaula throw
   $thia ·>lin• • —LI..-,   //dela clluaula throw
   $thia ·>trace  •cSebu9_backtracell
   $tbia ·>atring  • StriI19Format($tbial1


protected S•••age • 'Unlmovn exception•;  II • nrje de la excepci6a
protected $code • 01  II c6cligo de la axc:epci6a definida por el uauario
protected $file,       // D0llbre de archivo del origen de la excepc:1.6a
protected $1:1.ne;       II Unea  del origen de la excepci6a

private $trace;  // piata da la excepci6a
private $atring; // a6lo interno

final fuactioa getNeaaageO (
   return $thia ·>-••ag~,
}
final fuactioa getCode()
   raturn $thia•>code1.

final fuaction getFile11
   return $thia ·>file;
}
Unal  fwac:tion getTrace(1  (
   return $thia•>trace1
}
final function get'l'raceAaString() (
   retura aelf::'l'raceFormat1$thial;
t
functioa _toStr:1.ng(l (
   retura fthia ·>atring;
}
atatic private fuactioa StringFoEWlllt(Sxcaption $exception1  (
   II ''. una funci6a DO diapoaibla en aecueaciaa de e-do•     de PBP
   // que devuelve toda la info:rmaci6a relevance ea forma de cadena .
|
•tatic private fuactioa 'l'racePormat(Bxceptioa $excaptioal  (
   II .'. una funci6a DO diapoaible ea Hcueaciaa de -.adoa     de PBP
   II que devuelve la piata como cadena

|
? >
```

La razón del ~n61isisde esta definición declue es que la mayorfa de los dtod~ públicos son definitivos, no se pueden reemplazar. Puede erar su propia subd..,e' Bxception pero no puede modificar el comportamiento de los mftodos b4sicos. Fíjese en que puede reemplazar la funcidn_tOString (), pua modificar la iorma de representar la excepción. Tambim puede wdir sus propios mftodos. En el listado 7.3 se recoge ~ ejemplo de clase Bxception definida por el usuario.

EL_HOLISTICO@HOTMAIL.COM - 2006

Uamdo 7.3. user_deflned_exception.php. Ejemplo de c1... Exc:eptlon definida par el USUIlrio.

```
c?php

el••• aySacaption axtenda !xcaption
(
   function _ toString(l
   (
      return •ctable bordar>ctr>ctd>catrOD9>Bxception '  $thia->9etCode()
               •c/atrong>: '.  Sthia->9etMeaaage().'cbr  />'.'  in•
               Sthia->9etFile(l.  ' on line '  $thia->9etLine()
               'c/td>c/tr>c/teble>cbr  />';


try
(
   throw nev myBxceptionC'A  terrible  error baa occurred',  42l;
}
catch  (myBxception S•l
(
   echo S•l


?>
```

En este código declaramos una nueva clase de excepción, myixception, que ámplfa la clase Bxception b6sica. La diferencia entre esta d——— clase BXceptio? es que ieemplazamos "eriñefoao... tostrlngTf para poder imprimir la excepción de forma correcta. En la figura 7.2 puede comprobar el resultado de ejecutar este código.

Fltura 7.2. La clase myExc:eption proporcionaexcepciones con una atractiva Impresl6n.

Este ejemplo es muy sencillo. En el siguiente apartado veremos cómo crear diferentes excepcione& para solucionar los distintos tipos de error.

Excepciones en el ejemplo Bob's Auto Parts

En un capítulo anterior vimos cómo se podían almacenar los datos de los pedidos de Bob en un archivo plano. Sabemos que la E/S de datos (en realidad, cualquier tipo de E/5) es una parte de los programas en la que se generan numerosos errores, motivo por lo que es el candidato perfecto para aplicar el control de excepciones.

Si retomamos el código original, descubrimos tres aspectos que pueden dar problemas al escribir en el archivo: el archivo no se puede abrir, no se puede obtener un bloqueo o no se puede escribir en el archivo. Hemos creado una clase de excepción para cada una de estas posibilidades. En el listado 7.4 se reproduce el código de estas tres excepciones.

Llatado 7.4. flle_exc:eption.php. Excepciones rei.cionades con l..afS de d.tos.

```
c?php

el••• file<>penSaception extenda Bxception
(
   fnnction  ..... tDR1Fi.a9CI
   (
      return •fileOpenSxception '.  $thia->9etCode(l
              ' '.  '.  Sthia->9etMeaaage().'cbr  />'.'in'.
              . Stbia->9etFile(l.  ' on line '.  Stbia->getLine(l
              . •cbr  />' 1


claa• fileNrit.-XC:eption extenda Bxception
(
   function _ toString(I
   (
      return 'fileNriteBxception  '  Sthia->getCodell
               ', '.  $tbia->9etMeaaage().'cbr  />' .• in'
               Sthia->getFile().  ' on line '.  $thia->9etLine(l
               'cbr  />.


claa• fileLoclc&xception extenda Bxception
(
   function _ coString(l
   (
      return 'fileLoclclxcepcion '  Stbia->getCodeCI
              ' '.  '.  Sthia->9etMeaaage().'cbr  />'.'  in•
              ' $tbia->9etFile().  ' on line '.  Sthia->getLine(l
```

```
• 'cllr />ıl
```

Las lUbdM11 Bxc:eptian no meen nada especialmmte intenwtte. De hecho, en lo qae nspecta a esta aplk:acidn, podriamol dejllrlu vadas o utilizar la clase Bxceptian proporcionada. No obstamehemoe incluido un método _toStr.ing () para cada una qae explica el tipo de excepci6n geaerado.

Volvemos a escribir el archivo proceuorder.php de un capftulo anterior para incorporar las excepciones. La nueva versión se incluye en el listado *75*.

Uatado 7.5. praceuorder.php. Sec:uencllI de COffl8lldo9 de proc:eAfflientl de Bob con control de exc:epclones incluido.

```
c?pbp

    require_oace(ıfile~exe-,tiC11111,pbp'Il

    // c:zee .-re•    de YU'iabl•   corco•
    $tir91¡Cy  •  $_P081'(ıtir911Cy')1
    $oUqCy.    s_IPOa'(ı0Uqcy•1,
    S•pañqcy   •  $_IOIT(•epeñqey'l    ı
    Saddre••   •  $_IOIT['addre••'J;

    $DOCUMENT_ROOT = $_SERVER['DOCUMENT_ROOT'];
7>
cbtal>
cbe•ib
    ctitle>llelb.'• Auto ••re• - Order ... ult•c/titl•>
c/M•d>
é1icidy>
cbl>lloll'•  Auto P•rt8c/b1>
cb2~r        len1C.c/b2>
c?pbp
Sdet•  • ut•C  'a,1.  :lS r• ı

ecbo 'cp>c>zder proc••••    at '1
ec:bo Sdete,
ecbo 'c/p>';

echo '<p>Your order is as follows: </p>';

Stotelqey • 01
Stot•lqty  •StireqCy   + Soilqty   + $11parkqtyl
ec:lla 'lt-     ordered:  '.$tot•lqty. •cbr  />' ı

if(  Stot•lqty  ..  01
(
    ec:bo 'Toa did -t   order   eaytbilll!I aD tbe pre.-iou•   pegelcbr />';

•1••
(
```

```
u ℓ Stinqty>'o ›
    ecbo Stinqty.   • tireecbr    />' ı
if  ( $oil4CY>Ci
    ecllo $oUqcy.  • bottl••    of oUcbr   />ıl
if  I Sepu-lrqty>0)
    ecbo Sepazqcy.  • q,art   pl119•cbr  />';


Stot•l —— t  •  0.00,

defiM ℓ •n....acs•.  1001,
dllftM ℓ 'ou..a:i:cs ''  101;
dllfiM('SJIUDalCS',  4ll

Stot•l-t          • $tireqty    • TlUftJC:S
      + $o1lqty  • OJLNJ:CS
      + $11pañqty  • SPUDUCS1

$totalamount=number_format($totalamount, 2, '.', ' ');

ecbo 'cp>Total  of  order  i•  '.$totel-t. 'c/p>'  ı
ecbo 'cp>Addrea•  to  abip  to  i•  ',Seddr•••·  'c/p>'  ı

$aaéplltatrilll!I  •  $dllta. •\t•.ttinqcy. • úrea   \t•  .$oilqty.•  oÚ\t•
    .$epu-kqty.•  eperlc plaga\t\$•.$totel-.owat.
    .•\t •.  SIldlkeu. •\n•,

// abra  el  •rcbiYO  pu-•  ad:1-tU'   •1-toe
trr
(
    if (!($fp = @fopen("$DOCUMENT_ROOT/../orders/orders.txt", '&b'III
        throw new fileOpenException();

    u .. uuon•fp,  r.oac_au
        tbrow  uv  UleLackDceptiaaCI;

    u  llfvri.teCSfp,  $aatputatri11111,  etrl•C$Olltputetrillgll)
        t~     nev fil .. ritelsception.111
    Uoc:JcC$Cp,  LOCIC_ml  ı
    fcl-Ctfptl
    ecbo 'cp~          vritt•,c/p>';
)
cetcb   CfileOpenlsc.ptioa      $foel
(
    ecbo 'cp>catroag..ordera      file   CCNIdDOt be opeaed. '
        .•Pl .... contect   aur ....,.•ter       for llelp.c/at%Gllll>c/p>ıl
}
catcb   Clsc-,tioa     S•I
(
    ecllo  'cp>ca~Yoar          ordff   coald DOt be prace•aed at  tlaia  ti•.
        .'Pl-        tri,  ...  i.a  l•ter,c/~:i,c/p>•1


?>
c/'tlolly>
c/btml>
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Puede apreciar qué la sección de E/S del archivo se incluye en un bloque try. Es recomendable utilizar bloques try de pequefto tamaño y capturar las correspondientes excepciones al final de cada uno. De esta forma el código de control de excepciones resulta más sencillo de programar y mantener ya que en todo momento podemos ver lo que sucede.

Si no puede abrir el archivo, genere una excepción fileOpenBxception; si no puede bloquearlo, genere una excepción fileLockBxception y si no puede escribir en el archivo, genere una excepción filelfriteBxception.

Detengámonos en los bloques eaeen. Para ilustrar este concepto sólo hemos incluid~ dos: uno para controlar fileOpenBxception y otro que se encarga de Exceptions. Como el resto de excepciones se hereda de Bxception, se capturarán por medio del segundo bloquec:atcb. Los bloques catcbsecomparan siguiendo los mismos criterios que el operador inatanceof, un buen motivo para ampliar sus clases de excepción de una misma clase.

Una advertencia importante: si genera una excepción para la que no ha escrito el correspondiente bloque catcb, PHP generará un error muy grave.

Excepciones y otros mecanismos de control en errores de PHP

Además del mecanismo de control de excepciones que hemos analizado en este capitulo, PRP cuenta con una comple.,a compatibilidad con la resoluaon de errores, que veremos en un capítulo posterior. El proceso de generar y controlar excepciones no interfiere en absoluto con el funcionamiento de este mecanismo de **resoludóndeerrores**

En el listado 7.5 a la invocación de fopen (1 se ha añadido como prefijo el operador • de supresión de errores. Si faHa, PHP generará una advertencia que puede o no informarse o registrarse, en función de los parámetros de informe de errores configurados en pbp. ini. Estos parámetros se describirin detalladamente en un capítulo posterior pero debe saber que esta advertencia se emite independientemente de si se genera una excepción o no.

Lecturas adicionales

Como el control de excepciones es una de las novedades de PHPS, no hay mucho escrito sobre este tema. Sin embargo, sí abunda información búic:a al respecto. Sun cuenta con un completo curso práctico sobre las excepciones y por qué utilizarlas (escrito desde la penpectiva de Java, evidentemente), que puede encontrar en bttp://java.sun.c:om/docs/books/tutorial/esaential/except!ons/ definition.btml.

A continuación

La siguiente parte del libro está dedicada a MySQL Explicaremos cómo crear y completar una base de datos de MySQL, y aplicaremos lo aprendido sobre PHP pan que pueda acceder a su base de datos desde la Web.

EL_HOLISTICO@HOTMAIL.COM - 2006

Parte 11

Utilizar MySQL

8

Diseñar una base de dalos vveb

Ahora que ya se ha familiarizado con los fundamentos de PHP, vamos a ver cómo integrar un base de datos en nuestras secuencias de comandos. En un capítulo anterior comentamos las ventajas de utilizar una base de datos relacional en lugar de un archivo sin procesar:

- Los RDBMSbrindan un acceso más rápido a los datos que los archivos sin procesar.
- Los RDBMS permiten realizar consultas fácilmente para extraer conjuntos de datos que encajen con determinados criterios.
- Los RDBMS incorporan mecanismos para tratar con accesos simultáneos para no tener que preocupamos de esta tarea como programadores.
- Los RDBMS brindan acceso aleatorio a los datos.
- Los RDBMS incorporan un sistema de privilegios.

Para ser más concretos, el uso de una base de datos relacional nos permite responder de forma rápida y sendlla a consultas sobre la procedencia de los clientes, los productos que se están vendiendo mejor o qué tipo de clientes son los que más dinero se dejan. Esta información puede ayudarle a mejorar el sitio para atraer y fidelizar a los clientes. En esta sección utilizaremos MySQLcomo base de datos.

Antes de profundizar en el estudio de las características de MySQL en el siguiente capítulo, examinaremos los siguientes aspectos:

EL_HOLISTICO@HOTMAIL.COM - 2006

- Conceptos y terminología relacionados con las bases de datos relacionales
- Oisei\o de bases de datos Web
- Arquitecturade bases de datos Web

En esta parte del. libro, nos centraremos en los siguientesaspectos:

- Creu una bae de datas: Analizaremos la configuración básica necesaria para conectar una base de datos de MySQL a la Web. Aprenderemos a crear usuarios, bases de datos, tablas e índices, así como los distíntos motores de almacenamiento de MySQL
- Trabaja con una bue de datos de MySQL: Veremos cómo consultar la base de datos y cómo ai\adir, eliminar y actualizarregistros, todo desde la línea de comandos.
- Acceder a una bae de datos de MySQL deecle la Web con PHP: Explicaremos cómo conectar PHP y MySQL para poder utilizar y administrar la base de datos desde una interfaz Web. Analizaremos dos métodospara ello: utilizar la biblioteca MySQL de PHP y el nivel de abstracción de base de datos PEAR:08.
- Administración avanzada de MySQL: Describiremos cómo administrar MySQL, con detalles sobre el sistema de privilegios, la seguridad y la optimización.
- Yrográmád6ñ-avmzadade MjSQL: Ana1izaremos los li\ótóres de almacenamiento, con especial atención en las transacciones, búsqueda de texto y los procedimientosalmacenados.

Conceptos de base de datos relacionales

Las bases de datos relacionales son, con diferencia, el tipo de base de datos mú utilizado. Estas bases de datos se basan teóricamente en el álgebra relacional. No es necesario comprenderla teoría para utilizar una base de datos relacional (aunque '\"endría bien), pero sí conviene comprender algunos conceptos básicos relacionados con las bases de datos.

Tablas

Las bases de datos relacionales se componen de relaciones, conocidas de manera más COD\ún como tablas. Una tabla es exactamente eso, una tabla de datos. Un ejemplo de tabla relacional es una hoja de cálculo electrónica. Vamos a examinar un ejemplo. Sil la figura 8.1, puede ver una tabla de ejemplo. Esta tabla contiene los nombres y las direcciones de los clientes de una librería, Book-0-Rama.

CUSTOMERS

a-o	.....	MRs.	ca,
1	.Ml9Smill	250lk	AlrparlWIII
2	MnWIDrig	1147HIIIIN.-...	lllll ..
3	lIldl9l9 Mu	317Nar111Amd	'_''''''

Flgwa 1.1. Loa detalles de loa clientes de Boak-0-Rama ae guardan en una tabla.

La tabla tiene un nombre (Customera), varias columnas que hacen referencia a un tipo diferente de datos y filas que se corresponden con los distíntos clientes.

Columnas

Cada columna de la tabla tiene un nombre exclusivo y diferentes datos. Cada una de ellas lleva asociado un tipo de datos. Por ejemplo, en la tabla CUstomers de la figura 8.1, puede ver que el campo CUstomerID es de tipo entero y las otras tres columnas son cadenas. Las columnas a veces se llaman campos o atributos.

Filas

Cada fila de la tabla representa un cliente distínto. Debido al formato tabular, todos tienen los mismos atributos. Las filas también se denominan registros.

Valores

Cada fila se compone de un conjunto de valo~ individualas qu, se. COl'J"-5P.9n':' den con las columnas. Cada valor debe tener el tipo de dato especificado por su columna.

Claves

Necesitamos disponer de una forma de identificar a cada di.ente. Los nombres no son una buena opción (si tiene un nombre muy utilizado, entenderá por qué). Tomemos como ejemplo a Julie Smith de la tabla customers. Si abre una guía de teléfonos de los EEUU, descubrirá una gran cantidad de entradas con ese nombre.

Podemos distinguir a Julie de diferentes formas. Es probable que sea la única Julie Smith que viva en la dirección que se indica. Sin embargo, el uso del nombre y la dirección en conjunto para distínguir a alguien resulta muy pesado y suena un poco a lenguaje legal. Además se necesita utilizar más de una columna de la tabla.

En este ejemplo, lo que hemos hecho, y es probable que se decante por bl misma solución en sus aplicaciones, es asignar un número de identificación de diente exclusivo (CustomerID). Se trata del mismo principio seguido para uignar números de cuenta o número de socios únicos. Esta técnica facilita el almacenamiento de los detalles en una base de datos. Un número de identificación asignado

EL_HOLISTICO@HOTMAIL.COM - 2006

artificialmente garantiza su exclusividad. Existen muy pocos tipos de información real que dispongan de esta propiedad, aunque se utilicen en una combinación.

La columna de identificación de una tabla se denomina clave o clave principal. Lu claves pueden constar de varias columnas. &te seria el caso si optamos por utilizar "Julie Smith del 25 de Oak Street, Airport West" para hacer referencia a Julie. La clave estaría formada por lu columnu Nombre, Dirección y Ciudad, pero su exclusividad no estaría garantizada.

Las bases de datos suelen constar de varias tablas y utilizan una dave para asociar una tabla a otra. En la figura 8.2, hemos agregado una segunda tabla a la base de datos. En esta tabla se almacenan los pedidos realizados por los clientes. Cada fila de la tabla Orden representa un único pedido realizado por un único diente. Sabemos quién es el cliente porque almacenamos su CUstomerID. Por ejemplo, si examinamos el pedido con 0rderID 2, descubriremos que lo ha realizado el cliente con CUstomerID 1. A continuación, si examinamos la tabla CUatomera, veremos que el cliente con CUatomerID 1 es Julie Smith.

CUSTOMERS

~	.....	Addlwa	Clly
	1.U.lllliil'l	2SOlk._	Alrpartw..t
	2 Alml'lWang	1147"--.-....	.....
	3MIdlll9Nlu	357Nallll11Md	V.....

OROE~

OldlllO	c.....IO	~	o.e.
1	3	27.50	m-Apr-2000
2	1	12.11	1~
3	2	74.00	1~
4	4	...	01.....,.....,

Figura 8.2. Cada pedido de la tablll Orders hace referencia a un diente de la tabla Customer.

El término utilizado en las bues relacionales para designar esta relación es clave secundaria. La columna CuatomerID es la clave principal de la tabla CUato—r, pero cuando aparece en otra tabla, como la tabla Orders, se conoce como clave secundaria.

Es posible que se esté preguntando por qué hemos optado por utilizar dos tablas distintas y no almacenar la dirección de Julie en la tabla Orden. En la siguiente sección intentaremos dar respuesta a esta cuestión.

Esquemas

El conjunto completo de diseños de tabla de un base de datos se conoce como esquema de la base de datos. Es como el esqueleto de la base de datos. Un esquema debe mostrar las tablas con sus columnas, los tipos de datos de lu columnas e

indicar la clave principal de cada tabla y sus claves secundarias. Los esquemas no incluyen datos, pero se puede incluir un ejemplo para explicar su objetivo. Los esquemas pueden adoptar la forma de diagramas, como los que estamos utilizando, diagramas de relaciones de entidad (que no se analizarút en este libro) o pueden tener forma de texto, como

```
euacaller•ICU•59!!rID, ..... lldrm••' Cityl
Order•(OrderID, CU•r—rZD, Am>uaC, Datel
```

Los términos subrayados con una lfnea continua en el esquema son las claves principales de la relación. Los términos en cursiva son lu claves secundarias de la reladón en la que aparecen.

Relaciones

Las claves secundarias representan una relación enae los datos de dos tablas. Por ejemplo, el vinculo de Orders a CUstomers representa una relación entre una fila de la tabla Orders y una fila de la tabla CUstomers.

Existen tres tipos búicos de relaciones en una base de datos relacional. Se clasifican en función del ndmero de elementos que haya a cada lado de la relación. Las relaáones puede ser uno a uno, uno a varios o varios a van:os.

Una relación uno a uno significa que sólo hay un elemento a cada lado de la relación. Por ejemplo, si hemos co_locado direcciones en una tabla distinta a.la de clientes, existiría una relación uno a uno enae ambu. Podrfamos tener 'IJIll cla.,!e secundaria desde la tabla de direcciones a la tabla de clientes o al revés (no serian obligatorias ambu).

En una relación uno a varios, una fila de una tabla se vincula a varias filas de otra tabla. En nuestro ejemplo, un cliente podrfa haber realizado varios pedidos. En estas relaciones, la tabla que contiene varias filas tendrá una clave secundaria huta la tabla con una sola fila. En este caso, hemos incluido la columna CustomerID dentro de la tabla Order para mostrar la reladón.

En las relaciones varios a varios, varias filas de una tabla se asocian a varias filas de otra tabla. Por ejemplo, si tememos dos tablas, Libro• y Autores, puede que un libro haya sido escrito por dos autores y que ambos hayan escrito otros libros, en solitario o en colaboración con otros autores. En este tipo de relaciones se. crea una tabla especial para representar las relaáones, con lo que nos tendremos las siguientes: Libros, Autores, Libros_Autores. La tercera tabla contendrá únicamente-' lu claves de las otras tablas como claves secundarias en pares, para mostrar qué autores están relacionados con cada libro.

Diseñar nuestra base de datos Web

La tarea mú difícil consiste en determinar cuútdo se necesita una tabla y cuál debería ser la clave. Éxiste una ingente cantidad de material dedicado a estudiar los

EL_HOLISTICO@HOTMAIL.COM - 2006

diagramas de reladón de entidades y de normálizadón de bases de datos, pero quedan fuera del ¸mbito.de este libro. Sin embargo, en la mayor parte de los casos, basta con seguir unos sencillos principios básicos de disei\o. Vamos a analizarlos en el contexto de la aplicación Book-0-Rama.

Pensar en los objetos del mundo real que se están modelando

Al crear una base de datos, lo que hacemos es modelar elementos y relaciones del mundo real, y almacenar la información sobre dichos objeto9 y relaciones. Por regla general, cada clase de objeto del mundo real que se modele necesitan su propia tabla. Piense en ello un momento: queremos almacenar la misma inlormadón sobre todos nuestros clientes. Si existe un conjunto de datos con la misma "fotma", podremos crear t,cilmente una tabla que se corresponda con dichos datos. En el ejemplo Book-0-Rama, queremos almacenar la información sobre nuestros clientes, los libros que vendemos y los detalles de los pedidos. Todos los clientes tienen un nombre y una dirección. Los pedidos tienen una fecha y una cantidad total, así como los libros pedidos. Los libros constan de ISBN, de un autor, un título y un precio. Por lo tanto, parece que necesitamos al menos tres tablas en esta base de datos: CUstomers, Orders y Books. En la figura 8.3 se recoge el esquema inicial.

CUSTOMERS

	·····	MdNs"
1	Julle Smllh	250.SIINI
2	AlmllWang	1147t.in.M9nlle
3	MIdIII9 Arllu	357 Nao1ll AaN

~ROE~

o..i.t>	~	Anad	o
1	3	27.50	02.....-.,
2	1	12.•	1~
3	2	74.00	1Mjlr.aJDO
4	4	●●	01.u...aJIID

BOOKS

ISIIN			
0472.31111.			
0472-3174$-1	ThanmDewn	....)... Oebilll GNIM.hix	24.9t
0472-3150N	PlUia, - ...	T- 'Ibu!NIIGWh 24 Haura	24.9t

Flguni 8.3. El esquema inicial consta de las tablas Customers, Orders y Books.

P~ el momento, no podemos determinar qué libros se incluyen en cada pedido a partir del modelo. Lo veremos en un instante.

Evitar el almacenamiento de datos redundantes

En una sección anterior ya nos planteamos por qué no almacenar la dilección de Julie Smith en la tabla de pedidos.

Si Julie realiza varios pedidos a Book-0-Rama, como así esperamos, acabaremos almacenando sus datos varias veces. Es probable que la tabla de pedidos presente un aspecto parecido al ilustrado en la figura 8.4.

OROERS

OIdIltO	Anad	O.	eu..n.tD	N9m	~	CIly
12	1 ▬.ao	25-Apr,,21DJ	1	Julia Smllh	21 OIII SIINt	~W.
13	43.00	~	1	Julia Smill	21 OIII SIINt	~W.
14	15.11	'''''''';	1	JulleSmill	21 OIII SIINt	~W.
15	23.75	01-Mlp,2IIOC	1	Julia Smill	21 OIII SIINt	~W.

Figura 8.4. El disel'lode una base de datos que almacena datos redundantes aumenta el espacio utilizado y puede dar lugar a anomallasen los datos.

Esto plantea dos problemas básicos.

- El primero supone malgastar espacio innecesariamente. ¿Por qué guardar los detalles de Julie en tres sitios si sólo necesitamos guardarlos en uno?
- El segundo problema son las anomalías que pueden surgir al actualizar los datos, es decir, situaciones en las que al cambia(los datos se producen inconsistencias. La integridad de los datos quedará alterada y ya no sabremos qué datos son correctos y qué datos son incorrectos. Por regla general, esta situad~d9:r, lugar a ~ ~~dida de datos.

Es necesario evitar tres tipos de anomalías de actualización: anomalías de modificación, de inserción y de eliminación.

Si Julie se muda de domicilio con pedidos todavía pendientes, tendremos que actualizar su dirección en tres lugares en lugar de en uno, lo que equivale a tener que realizar el trabajo tres veces. Resulta sencillo pasar por alto esta tarea y acabar realizando el cambio en un solo lugar, lo que dan lugar a la incoherencia de los datos de la base de datos (algo muy negativo). Estos problemas se conocen como anomalías de modificación porque tienen lugar cuanto se intenta modificar la base de datos.

En este disetlo, necesitamos insertar los detalles de Julie cada vez que recibimos un pedido, lo que significa que deberemos comprobar y aseguramos de que sus datos son correctos en las filas existentes de la tabla. Si pasáramos por alto la comprobación, podríamos acabar con dos filas de datos desiguales sobre Julie. Por ejemplo, una fila podría indicar que vive en Airport West y otra en Airport. Este tipo de anomalía se conoce como anomalía de inserción porque tiene lugar al introducir los datos.

El tercer tipo de anomalía se conoce como anomalía de eliminadón porque tiene lugar (sorpresa, sorpresa) al eliminar filas de la base de datos. Por ejemplo, ima~e

EL_HOLISTICO@HOTMAIL.COM - 2006

que tras enviar un pedido, lo eliminamos de la base de datos. Cuando se hayan servido todos los pedidos de Julie. se habrán eliminado todos de la tabla Orders. Esto significa que no dispondremos de un registro sobre la dirección de Julie. Por tanto, no podremos enviarles ninguna oferta especial y necesitaremos tomar sus datos de nuevo cuando vuelva a realizar un pedido. Por regla general. al disdar las bases de datos se intenta evitar estas anomaUas.

Utilizar valores de columna atómicos

Cada atributo de una fila debe almacenar un solo valor. Por ejemplo, necesitamos saber qué libros componen cada pedido. Existen varias formas de hacerlo.

Podemos agregar una columna a la tabla Orders en la que se enumeren todos los libros que hemos pedido. como se ilustra en la figura 8.5.

ORDERS

OnlIIIII	~	~	Oll9	IlooaOnlnd
1	3	27.!0	0z....zaa0	CMI72-31117-I
2	1	12.99	15-Apr-2111111	IMI72-317~1.CMI72-3111111-2
3	2	74.00	1Mp,-2DIIO	0472-31117-1
4	3	UI	01.....,.aaao	IMI72-317~1. 0472-315CJI.ZQ.472-31117-1

FIguni 1.5. En este disetlo, el atributo Boolcs Orclered de cada lila incluye varloa valores.

No es una buena idea por varias razones. Lo que en realidad estamos hadendo es anidar una tabla dentro de una columna (una tabla que relaciona pedidos y libros). Al hacerlo, resulta mú dificil responder a preguntas como •¿Cuántas copias del libro /aw, 2fo, *Profassioal lhwlopm* quiere pedir?" ya que el sistema no puede contar los campos coincidentes. En su lugar, debe analizar cada valor de atributo y determinarsi contiene una correspondencia en su interior.

Como lo que estamos haciendo es creu una tabla dentro de otra. lo que deberiamos hacer es crear una nueva tabla. Esta nueva tabla, se denomina Order _I tema y se muestra en la figura 8.6.

OROER_ITEMS

OdlrtD	ISIIN	QumIlly
1	CMI72-31.7-I	1
2	0472-317—1	2
2	0472-3111111-2	1
3	CMI72-31N7-I	1
4	0472-3174~1	1
4	0472-3111111-2	2
4	0472-31117-1	1

FIgu ... 1.1. Eata dlsello facilita la IM)squeda de libros concretos pedidos.

Esta tabla proporciona un vínculo entre la tabla Orders y la tabla Booka. Es habitual crear este tipo de tablas en una relación varios a varios entre dos objetos:

en este caso, un pedido puede constar de varios libros y varias personas pueden pedir un libro.

Seleccionar claves lógicas

Asegúrese de que las claves seleccionadas son exclusivas. En nuestlO caso, hemos creado una clave especial para los clientes (CUst011erID) y para los pedidos (OrderID) porque es posible que estos objetos del mundo real no dispongan de manera natural de un identificador con garantías de exclusividad. No necesitamos crear un identificador único para los libros porque ya existe en forma de ISBN. En Order_Item, puede agregar una clave adicional si lo desea, pero la combinación de los atributos OrderID e ISBN resultará exclusiva siempre y cuando se trate a cada copia de un libro dentro de un pedido como una fila. Por resta razón, la tabla Order_Itea incluye una columna Quanti ty.

Reflexionar sobre las preguntas que desea formular a la base de datos

Enlazando con la sección anterior, piense en las preguntas que illsea que responda la base de datos. (Recuerde las preguntas que realizamos al principio del capitulo. Por ejemplo, ¿cuáles son los libros que más vende Book-0-——?) Asegúrese de que la base de datos contiene todos los datos necesarios -r que se han establecido los vínculos adecuados entre las tablas para responder aJas preguntas deseadas.

Evitar diseños con varios atributos vacíos

Si desea agregar reseñas de libros a la base de datos, dispone al menos dos formas de hacerlo. En la figura *8.7* se recogen ambos enfoques.

BOOKS

191	,,,,,,,,,,	ni.	.....	~
0472-31117-1	Mldml	... 21ar ..... ~	34.91	
Q.472-317~1	lllonmDown	..... 0--, GNUÜIUX	24.99	
0472-3111111-2	.....	'IWdl 'fblnIll GM' .. 24 Haunl	24.99	

BOOK REVIEWS

FIgu... 1.7. Para agregar resetlas, podemos incorporar una nueva columna a la tabla Bookl o al'ladlt una tabla apec:ltlca para almacenar 9118 infonnac:ión.

EL_HOLISTICO@HOTMAIL.COM - 2006

La primera comiste en agregar la colmnna Review a la tabla Booka. En esta solución se incluye un campo nuevo de reseña para cada libro. Si la base de datos consta de una gran cantidad de libros y el encargado de las reseñas no tiene previ,s. to crear una para cada libro, muchu filas quedarán ,•acfas. Es lo que se conoce como valor nulo. No conviene tener muchos valores nulos en una tabla ya que esta práctica contribuye a desperdiciar espado de almacenamiento y ocasiona problemas al calcular totales o al aplicar otras funciones a columnas numéricas. Si un usuario ve un valor nulo en una tabla, no sabrá si se debe a que el valor no resulta relevante, a que hay un error en la base de datos o a que no se han introducido los datos todavía. Por regla general, para evitar los problemas con la inclusión de demasiados valores nulos se recurre a un diseilo alternativo. En este caso, podemos utilizar el segundo diseilo ilustrado en la figura 8.7. En este caso, sólo se enumeran los libros con una reseila a la tabla Book_Reviews, asf como sus reseñas.

Ff¡ese en que este disefto se basa en la idea de tener una perso~ encargada de la revisión, es decir, una relación uno a uno entre las tablas Books y Reviews. Si deseara incluir diferentes resellas de un mismo libro, seria una relación uno a varios, y tendría que aplicar la segunda opción de diseilo. Con una resella por libro, podria utilizar el ISBN como clave principal de la tabla Book_Reviews. Si hubiera varias. reseñas por libro, debería ai\adir un identificador exclusivo para cada una.

Resumen de los tipos de tablas

CelRG tenck4-laoportunidad de comprobar, las bases de datos suelen componerse de dos tipos de tablas:

- tablas sencillas que describen un objeto del mundo real. También podrían coñtener claves a otros objetoiUimples en tos"que emta una relaci6rnmo a uno o uno a varios. Por ejemplo, nuestro cliente podría contener varios pedidos, pero cada pedido se asocia a un único cliente. Por lo tanto, asignamos una referencia al cliente en el pedido.
- Tablas de unión que describen una relación varios a varios entre dos objetos reales como la relación entre Orders y Books. Estas tablas suelen asociane con algún tipo de transacción del mundo real.

Arquitectura de base de datos Web

Tras comentar la arquitectura illtema de nuestra base de datos, vamos a examinar la arquitectura externa de un sistema de base de datos Web y a comentar la metodología para desarrollar un sistema de base de datos Web.

Arquitectura

En la figura 8.8 se ilustra el funcionamiento básico de un servidor Web. Este sistema consta de dos objetos: un navegador Web y un servidor Web. Para unirlos

se necesita una forma de comunicación. Un navegador Web realiza una solicitud al servidor. El servidor devuelve una respuesta. Esta arquitectura resulta apropiada para un servidor que devuelva páginas esblticas. La arquitectura que devuelve resultados de un sitio Web basado en una base de datos resulta un tanto m6s complicada.

Flgun 8.8. La ,alaci6n cllents/Hf'lldor entre un 1111vepdor Web y un Nrvldor Web requiere comunlcllcl6n.

Las aplicaciones de base de datos Web que desarrollaremos en este libro siguen la estructura general de base de datos Web que se ilustra en la figura 8.9. Gran parte de esta estructura debería resultarle familiar a estas alturas.

Flgu,. 8.9.La arquitectura búica de base de datos Web se compone de uñ navegador Web, un servidor Web, un motor de HCuenc:111 de CO!Mndos y un servidor de I!ue de datos.

Una transaa:ión típica de base de datos Web se compone de las fues siguientes, numeradas en función de la figura 8.9. Examinaremos estas fases en el contexto del ejemplo Book-0-Rama.

1. El navegador Web de un usuario envía una petición~ a una página Web dada. Por ejemplo, puede tratarse de una búsqueda para extraer todos los libros escritos por Laura Thomson en Book-0-Rama con ayuda de un formulario HTML La página de resultados de búsqueda se denomina resulte . php.
2. El servidor Web recibe la petición de reaul te. php, recupera el archivo y lo pasa al motor de PHP para su procesamiento.
3. El motor de PHP comienza a analizar la secuencia de comandos. Dentro de la secuencia de comandos hay un comando que establece la conexión a la base de datos y ejecuta una consulta (realiza la búsqueda de libros). PHP abre una conexión al servidor MySQL y remite la consulta pertinente.
4. El servidor MySQL recibe la consulta de la base de datos y la procesa. A continuación, envíli. los resultados (una lista de libros) al motor de PHP.
5. El motor de PHP termina de ejecutar la secuencia de comandos, lo que suele implicar la aplicación de formato a los resultados en HTML Seguidamente, devuelve el código HTML resultante al servidor Web.
6. El servidor Web devuelve el código HTML al navegador dopcie el usuario puede ver la lista de los libros solicitados.

EL_HOLISTICO@HOTMAIL.COM - 2006

El proceso• Wliamente el mismo independientemente del motor de NCIMIIICia de commdos o del senidor de base de datos q• utilicemos. Por regla general, el software del servidor Web, el motor de PHP y el semdor de la bMe de datos se ejecutan en el mismo equipo. Sin embargo, tambim es habitual ejecutu el semdor de la bue de datas en un equipo difmente, por razones de seguridad, de una mayor capaddad o de reparto de la carga. Desde el punto de vista de desarrollo, el trabajo ser6 el mismo, pero puede pnsentu aJgunu ventaju significativas de rendimiento. Al aumentar el tamai\o y la complejidad de las apllcacioMs, tendr6 que separar sus aplicacioaes PHP en niveles, por lo general un nivel de bue de datos que interactúe con MySQL, un nivel de 16gica de negocios con las~ báicu de la aplic:ación y un nivel de ——— 6n que gestione el n11ultado HlML. Sin eatbargo, la arquitectura búica ilustrada en la figura 8.9 es dlida; basta con dadir estructura a la sec:dcm de PHP.

Lecturas adicionales

En este capitulo, hemos analizado alguna de las direclrices para el disefto de bases de datos. Si quiere pmfanctizar en la teoría sobre la que dacanaan las bues de j,datoarebdonales, puede leer algunas de las obras escritas por expertos en el tema C01110 C.J. Date. Eso si. tenga en cuenta que el material puede resultar bastante teórico y poco relevmte a corto plazo para un desurollador Web comen:iaL Las bases de datos Web más habituales no suelen alanzar un nivel IIIn alto.de complejidad.

A continuación

En el siguiente capítulo, comenzaremos a configurar nuestra base de datos MySQL En primer lugar apnnden a configurar una bue-de datos MySQL para la Web)"!t6alo consultarla y, a continuación. c:6mo COJllllltarla desde PHP.

EL_HOLISTICO@HOTMAIL.COM - 2006

9

Crear la base de datos Web

En este capítulo vamos a comentar cómo configurar una base MySQL para su uso en un sitio Web.

Abordaremos los siguientes aspectos:

- Crearunabasededatos
- Definir usuarios y privilegios
- Introducción al sistema de privilegios
- Crear tablas de base de datos
- Crear índices
- Seleccionar tipos de columna en MySQL

En este capítulo, vamos a seguir adelante con la aplicación de la librería en línea Book-O·Ramacomentada en el último capítulo. A continuación, se recoge el esquema de la aplicación de Book-0-Rama para ayudarle a recordar:

```
Cu•t01111r•ICU•t01111rID,  ......  Addr•••, Cityl
order•lorderID, CU•t01111rlD, Amount, Datel
Book•IIsiii.'"""iiitbor, Title, Pricel
order__工ヒ__丄～. lSM,  Qua.ntityl
Book_Revieve(lSaN, a.. ievel
```

Recuerde que lu cla,es principales aparecen subrayadas con una línea continua y las claves secundario se representan en cursiva.

EL_HOLISTICO@HOTMAIL.COM - 2006

Para utilizar el 111Aterial de esta sección, debe disponer de acceso a MySQL, lo cual significa que

- Ha completado la instalación básica de MySQL en su servidor Web. Esto incluye:
  - Instalar los archivos
  - Configurar un usuario para ejecutar MySQL
  - Configurar su ruta
  - Ejecutar mysql_install_db,si resultara necesario'
  - Establecer la configuración del usuario raíz
  - Eliminar el usuario anónimo y la base de datos de prueba
  - Iniciar el servidor de MySQL por primera vez y configurarlo para que se ejecute de forma automática
- Si ha realizado todas estas operaciones, puede seguir adelante con ia lectura del capitulo. Si no lo ha hecho, en el apéndice A encontrará instrucciones sobre cómo proceder.

  Si tiene problemas durante este capítulo, puede que se deba a la configuración de su sistema MySQL En este caso, revise esta lista y el apéndice A para asegurarse de que la configuraciónes correcta.
- Dispone de acceso a MySQL en un equipo de cuya administración no se encarga, como un servido de alojamiento Web, un equipo en su lugar de trabajo, etc.

  Si éste fuera el caso, para poder seguir los ejemplos o crear su propia base de datos, deberá pedir a su administrador que configure un usuario y la base de datos para permitirle trabajar con ella y que le indique el nombre de usuario, la contraseña y el nombre de la base de datos que se le ha asignado.

  Puede ignorar las secciones de este capitulo dedicadas a explicar cómo configurar usuarios y bases de datos o leerlas para explicar mejor lo que necesita su administrador del sistema. Como usuario normal, no podlá ejecutar los comandos para crear usuarios y bases de datos.

Los ejemplos de este capitulo se han elaborado y probado con la última ¸·ersión 5.0 de MySQL Las versiones anteriores disponen de menor funcionalidad. Debería instalar o actualizar su versión a la más reciente, que puede descargar del sitio de MySQL en http://mysql.com.

En ~te libro interactuaremos con MySQL por medio de un cliente de lfnea de comandos denominado monitor de MySQL, que se incluye en todas las instalaciones de MySQL No obstante, puede utilizar cualquier otro cliente. Si utiliza MySQL en un entorno Web alojado, los administradores de sistemas suelen ofrecer la interfaz phpMyAdmin. basada en un navegador. Los distintos clientes GUI implican el uso

de procedimientos düerentes a los que describiremos en este capítulo pero no le resultan complicado adaptarlos a su caso c:onaeto.

Utilizar el monitor de MySQL

En este capítulo y en el siguiente, apreciará que en los ejemplos de MySQL cada comando termina en un punto y coma (;). Este símbolo indica a MySQL que ejecute el comando. Si olvida utilizar el punto y coma. no se aplicará ninguna acción. Se trata de un problema común entre los nuevos usuarios.

Tambim significa que puede utilizar nuevas líneu en mitad de un comando. En el libro, hemos utilizado este recurso para facilitar la lectura de los ejemplos. Apreciará dónde lo hemos hecho porque MySQL incluye un símbolo de continuación. Se trata de una flecha como la incluida en este fragmento:

```
ay•ql> grant ••l•et
    ·>
```

Este símbolo indica que MySQL espera más entradas. !"fientrasno se escriba un punto y coma, obtendrá estos caracteres al pulsar Jntro.

Tenga en cuenta ademú que las instrucciones de SQL no discriminan entre mayúsculas y minúsculas, pero que las bases ele datos y las tablas sí pueden hacerlo (este aspecto se ampliará mú adelante).

Iniciar sesión en MySQL

Para iniciar sesión, diríjase hasta la interfaz de línea de comandos de su equipo y escriba lo siguiente:

```
mysql -h nombredehost -u nombredeusuario -p
```

El comando myeql invoca al monitor de MySQL. Se trata de un cliente de línea de comandos que nos permite comunicarnos con el servidor MySQL.

El modificador -b se utiliza para especificar el host al que establecer la conexión, es decir, el equipo en el que se está ejecutando el servidor MySQL. Si ejecuta este comando en el mismo equipo que el servidor MySQL, no necesitará utilizar el modificador -h ni el parámetro nollbredebo•t. De lo contrario, deberá sustituir dicho par6metro por el nombre del equipo en el que se esté ejecutando el servidor MySQL

El modificador -u se utiliza para especificar el notlbredeusuario con el que desea conectarse. Si no se especifica, se utilizará el nombre de usuario predeterminado con el que se inició la sesión en el sistema operativo.

Si ha instalado MySQL en su propio equipo o servidor, deberá iniciar sesión como root y crear la base de datos que utilizaremos en esta sección. Si ha realiZJdo

EL_HOLISTICO@HOTMAIL.COM - 2006

una instalación limpia, el usuario raíz (root) ser6 el único usuario del que dispondr6 para iniciar la sesión. Si está utilizando MySQL en un equipo administrado por otra persona, utilice el nombre de usuario que se le haya asignado.

El modificador -p indic!I al servidor que se desea establecer la conexión utilizando una contraset\a. Puede evitar su uso, si no se ha establecido una contraseña para el usuario utilizado para iniciar la sesión.

Si se está iniciando la sesión como usuario raíz y no se ha establecido una contrasella para dicho usuario, le recomendamos que consulte el apéndice A de inmediato. Sin una contrasella para el usuario raíz , su sistema es inseguro.

No es necesario que incluya la contrasef\a en esta linea. El servidor MySQL se la pedir6. De hecho, es mejor no hacerlo. Si introduce la contrasella en la linea de comandos, se mostrar6 en la pantalla y otros usuarios podrían verla.

Tras introducir el comando anterior, debería obtener un resultado parecido al siguiente:

(Si no funcionase, verifique que el servidor MySQL se est6 ejecutando y el eerríando mysql aparece en algún lugar de su ruta.)

Debería introducir la contrasella. Si todo va bien, se mostraú una respuesta parecida a la siguiente:

```
lt~l-    e.o e.be NySOL —ic.or.  Co-«l•  •nd vic.b:  or \g.
-·-Yoe•r  M\<SQLcnnnectiAD  id 141 .5.Cg  au-v_.r:: _~UiQ!H  ~,U_,,i-l.c.
~    'belp;' or '\b' for belp. Type '\e' to clear e.he buffer.
,-Yllql>-
```

En su equipo, si no obtiene una respuesta similar, asegúrese de haber ejecutado my•ql_install_dh, si resultara necesario, de haber establecido la contraseita del usuario raíz y de haberla escrito correctamente. Si se trata de otro equipo, asegúrese de haber esaito correctamente la contrasella.

Tras ello, deberla aparecer el símbolo de comandos de MySQL U.to para aear la base de datos. Si est6 utilizando su propio equipo, siga las directrices de la siguiente sección, Si está utilizando el equipo de otra persona, esta fase ya estar6 completada. Puede saltar a la sección "Utilizar la base de datos correcta". Si lo desea, puede leer las secciones intermedias para ampliar sus conocimientos pero no podri ejecutar los comandos indicados en ellas. (O no debería, al menos).

Crear bases de datos y usuarios

El sistema de base de datos de MySQL admite una gran cantidad de bases de datm diferentes. Por regla general, se utilizar6 una bue de datos por cada aplicad6n. En nuestro ejemplo Book-0-Rama, la base de datos se llamar6 books.

Crear la base de datos

&ta es la parte más sencilla. Escriba la siguiente secuencia en el sfmbolo de comandos de MySQL:

```
mysql> create database nombrebd;
```

Sustituya el par6metro *runnlmbcl* por el nombre de la base de datos deseado. Para comenzar a crear el ejemplo Book-0-Rama, puede crear una base de datos llamada books.

Y eso es todo. Debería ver aparecer una respuesta como la siguiente:

```
Query OK, 1 rov affected 10.0, ••el
```

Esta secuencia indica que todo ha funcionado correctamente. Si no la obtiene, asegúrese de incluir el punto y coma al final de la línea. Estt sfmbolo indica a MySQL que ha terminado y debería ejecutar el comando.

Deflnir usuarios y privilegios

Un sistema MySQL puede tener muchos usuarios. Como norma debería utilizarse el. usua.r.in raíz. para labora administrativas únicamente, por razones de seg1lridad. Deber6 configurar una cuenta y una contraseña para cada usuario del sistema. No es necesario que sean las mismas utilizadas fuera de MySQL (por ejemplo, los nombres.de.ususio y contrasei'ias.de Unix o NT). Los mismos princip.ios se eplic8-'_ al usuario raíz. Conviene utilizar contraseitas diferentes para el sistema *y* para MySQL, en especial en el caso de la contrasella del usuario raíz.

No es obligatorio configurar contrasellas para los ~ríos, pero sí muy aconaejable. A la hora de configurar una base de datos Web, conviene establecer al menos un usuario por aplicación Web. Es probable que se esté preguntado por qué. La respuesta reside en los privilegios.

Introducción al sistema de privilegios de MySQL

Una de las mejores funciones de MySQL es su avanzado sistema de privilegios. Un privilegio es el derecho a realizar una acdón dada sobre un determinado objeto y se asocia a un usuario concreto. El concepto es muy similar a loe permisos de archivo. Al crear un usuario dentro de MySQL. se le conceden un conjunto de privilegios en IQS que se especifica lo que puede y no puede hacer dentro del sistema.

EL_HOLISTICO@HOTMAIL.COM - 2006

Principio de asignación del privilegio más bajo

Este principio se puede utilizar para mejorar la seguridad de cualquier sistema informático. Se trata de un principio elemental pero muy importante que se suele pasar por alto a menudo. Su contenido es el siguiente:

> Un usuario (o proceso) deberla disponer del nivel más bajo de privilegio necesario para realizar la tarea que se le asigne.

Su aplicación debe ir más a114 de MySQL. Por ejemplo, pm_a ejecutar consultas desde la Web, un usuario no necesita disponer de todos los privilegios asociados al usuario raíz. Por lo tanto, deberíamos crear otro usuario que sólo disponga de los privilegios necesarios para acceder a la base de datos que acabamos de aear.

Configurar usuarios: el comando GRANT

Los comandos GRANT y JlEVOKE se utilizan para conceder y retirar los derechos a los usuarios de MySQL en cuatro niveles de privilegio. Estos niveles son:

- Global
- Base de datos
- Tabla
- Columna

E"n un instante veremos cómo se aplica cada uno de ellos. El comando GRANT" se utiliza para crear usuarios y concederles privilegios. La sintaxis general del comando GRANTes la siguiente:

```
02AIIT privilewia.    (col....,••I
0• ele-aco
TO nombre usuario !IDBlft'IFISD 3Y •eoacra•eJla•J
(UQUIU 'c;pciOIIH_HI)
(VITB [OaAIIT OPTIOJI | opcioaee_l!miceJ I
```

Las cl¸usulas incluidas entre corchetes son opcionales. La sintaxis de este comando incluye varios marcadores de posiáón. En primer lugar, *priflilegios,* equivale a una lista de privilegios separados por comas. MySQL consta de un conjunto definido de ellos, que se describen en la siguiente secáón.

El marcador de posición colurnfllls es opáonal. Puede utilizarlo para especificar privilegios a cada columna. Puede utilizar el nombre de una sola columna o una lista de.nombres de columna separadas por comas.

El marcador de posiáón, *tlnvnto* es la base de datos o tabla a la que se aplican los nuevos privilegios.

Puede CODCeder privilegios a todas las bues de datos especificando • • • como *tltmmto.* Es lo que se conoce como conceder privilegios globales. Tambiin puede

hacerlo especificando • únicamente si no va a utilizar ninguna base de dalm en concreto. Por regla general, especi.ficar4todas las tablas de la base de dalos camo nomoreod. •, una sola tabla de base de datos como nombreod. nOllbretabla o columnas específicas indicando nomoreod.nomoretabla y las colwnnas desadas en el marcador de posición *columnas.* Si est6 utilizando una base de datas especifica al remitir este comando y sólo se utiliza nomoretabla, se interpretar6 como una tabla de la base de datos actual.

El marcador de posición *nornbrt_uslllrio* sed el nombre del usuario con el que desea registrarse en MySQL Recuerde que no tiene por qui ser el mismo que el nombre de registro del sistema. Este padmetro también puede contener un nombre de host. Puede utilizarlo para distinguir, por ejemplo, entre Laura (interpretado como laura•hostlocal) y laura~trositio. com. Esta opáón resulta bastante útil porque los usuarios pertenecientes a diferentes dominios suelen utilizar los mismos nombres. Tambiffl permite incrementar la seguridad porque se puede especificar desde dónde se conectan los usuarios e incluso a qué tablas y datos pueden acceder desde una ubicaáón dada.

El marcador de posición *contrastr'fll* es la contraseila que deseamos utilizar para iniciar la sesión. Se aplican las reglas habituales para la selecáón de ~traseñas. E"n una sección posterior ampliaremos el tema de la seguridad, pero las contrasei\asno deben resultar sencillas de adivinar. Es aconsejable que contengan un combinación de caracteres en mayúsculas y minúsculas y caracteres no alfabéticos.

La c1'usula RBQOIRB le permite especificar que el usuario debe conectarse a través de SSL, así como otras opciones SSL. En el manual de MySQL encontrad mú informaáón sobre conexiones SSL a MySQL

La opción WITH GRANT OPTIOH, si se especifica, permite que el usuario indicado delegue sus privilegios en otros usuarios.

Puede especificar la c1'usula WITH como

```
MAX_QUUIBS_PBR_ROOKa
```

o

```
MAX_UPDATES_PER_HOUR n
```

o

```
MAX_COIIIIBCTIOJrS_PD_HOlla a
```

Estas cl6usulas le permiten limitar el número de consultas, actualizaciones o conexiones por hora que puede realizar un usuario. Pueden resultar muy útiles para limitar la carga de un usuario concreto en sistemas compartidos.

Los privilegios se almacenan en dnco tablas del sistema, dentro de la base de datos mysql. Estas cinco tablas se denominan mysql. user, mysql. db, mysql .host, myaql. tables_privymyaql .columna_privy se relaáonan directamente con los cuatro niveles de privilegio mencionados anteriormente. Comó; alternativa a GRART, p.uede alterar estas tablas directamente, como se analizar' en un capítulo posterior.

Tipos y niveles de privilegio

Existen tres tipos búicos de privilegios en MySQL: los privilegi?5 apropiados para su concesión a los usuarios habituales, los privilegios apropiados para su concesión a los administradores y dos privilegios especiales. Se puede asignar cualquier privilegio a cualquier usuario, pero es aconsejable restringir los privilegios de administrador a los administradores, siguiendo el principio de conceder el ni~el nw bajo de privilegios. Debeda restringir los privilegios concedidos a los usuanos para las bases de datos y tablu que ~iten utilizar. No.d~beria conceder acceso a la base de datos myaql a nadie a excepción de a un administrlldor. En esta base de datos se almacenan todos los usuarios, contraseftas y otros datos. (En un capítulo posterior analizaremos esta base de datos.) Los p~vilegios p~ra el res~ de los usuarios deberían establecerse directamente en función de los tipos específi~os de comandos SQL y si un usuario puede ejecutarlos. En el siguiente capítulo analizaremos en detalle estos col1UU1dos. Por el momento, buta con la descripción conceptual indicada. En la tabla 9.1 se recogen estos privilegios. En la columna Se aplica a se enumeran los objetos a los se pueden conceder cada privilegio.

Tabla 1.1. Prtvffegiol para usuarios.

Privilegio	Se aplica a	Descripción
SILZCT	tablas, columnas	Permite a los usuarios salec:c:ionar filas (regis· Iros) de tablas.
IIISBR'i'	tablas; columnas	Permite a los usuarios insertar nuevas llas en. las tablas
OPDATB	tablas, columnas	Permite a los usuarios modificar valores de filas de tablas existentes.
DBLZTZ	tablas	Permite a los usuarios eliminar filas de tablas existentes.
INDBX	tablas	Permite a los usuarios crear y ellmln.- Indices sobre tablas concr.-..
AL':!:R	tablas	Permite a los usuarios alterar la estruc:lura de tablas exislentes ag1'9{1811do, por ejemplo, columnas, cambiando el nombre de las calumnas o de las tablas y modlllcando los Upos de datos de las columnas.
	bases de datos y tablas	Permite a los usuarios eliminar bases de datos o tablas. SI especlllcai una base de datw o una tabla dada en el colllmldo GRAIIT. los usuarios sólo pod'*' crear dlc:ha base de datos o mbla, lo que slgnifica que mnd'*' que ellmiMr1a primero.
DROP	bases de datos y tablas	Permite a los usuarios eliminar bases de datos o tablas.

La mayor parte de los privilegios para los usuarios normales son relativamente inofensivos en ~rminos de la seguridad del sistema. Se puede utilizar el privilegio ALTER para resolver el sistema de privilegios cambiando el nombre de lu tablu, pero los usuarios suelen utilizarlo bastante. La búsqueda de la seguridad consiste en hallar un equilibrio entre la usabilidad y la protección. Debe tomar una decisión con respecto al privilegio ALTER, pero debe saber que se suele conceder a los usuarios a menudo.

Además de los privilegios enumerados en la tabla 9.1, existe el privilegio RBFBRENCBS que no se utiliza y el privilegio GRAHT que se concede con el parámetro WITH GRANT OPTION en lugar de en la lista de *priflil,gios*. La tabla 9.2 muestra los privilegios apropiados para su concesión a los usuarios administrativos.

Tabla 9.2. Privilegios para administradores.

Privilegio	Descripción
CRBATB '§MPQRMIY-TAaL&a	Peaaitaa.un admiaisltador Jdlllzar la palabra clave TBMPORAaY en una lnsCrucció́n CRBATB TABLB.
FILB	Permite la lectura de datos en tablas desde an:hlvos y vlc:eV9n8.-
LOCIC TABLBS	Permite el uso explicito de una instrucción LOCX TABLBS.
PROCBss	Permite a un administrador ver los pn)CNOI de servidor y cerrartos.
RBLOAD	Permite a un administrador Wllver a cargar tablas di concesión y ellml11acĺón de privilegios, host, registros y tablas.
RBPLICATION CLIBNT	Permite utllizar SBOlf STATUS en lefYidores principales y secundarios di replicación. como veremos en un capítulo posterior.
RBPLICATIOlr SI.AVE	Permite la l9plicaci6n de todos los HfYldores secundarios en el servidor principal, como veremos en un capitulo postenor.
SHOW DATABASSS	Permite acceder a una lista de bases de datos por medio de una lnstruc:cíón SHOW DATAIIASBS. Sin este prtvlleglo, los usuarios Sólo pueden wr las bases de datos en las que tengan otros prlYileglos.

EL_HOLISTICO@HOTMAIL.COM - 2006

SRUTDONII	Permite a un adminlsndor c:emirel servidor MySQL.
SOHR	Permite a un administrador eliminar subprocesos pertenecientes a cualquier usuario.

Estos privilegios se pueden conceder a usuarios que no sean administradores pero debe estar muy atento si esti considerando dicha posibilidad. El privilegio FILB es un tanto diferente. La posibilidad de cargar datos desde archivos puede contribuir a ahorrar mucho tiempo al volv~r a introducir los datos en la base de datos. Sin embargo, la función de carga de archivos se puede utilizar para cargar cualquier archivo que puede ver el servidor MySQL, incluidas las bases de datos pertenecientesa otros usuarios y, potencialmente, los archivos de contraseñas. Tenga cuidado al conceder este privilegio u ofrezca la posibilidad de realizar la carga de los usuarios. Existen otlOS dos privilegios especiales,que se recogen en la tabla 9.3.

Figura 9.3. Prhrileglos eapeciale1.

PrMlcglo	Dosc:ipsor
ALL	Tambl6n.. puede eac:riblr ALL PRIVILBGBS en luger de ALL.
OSAGB	No concede ningún privilegio. Este pm,ilegio cnta un usuario y le pennit8 l'llgilnrM, pero no se le concede ningún Olro privilegio. Por regla general, la concesión de olnJI pr1vilegioa se aulle dejar para un momento posterior.

El comando REVOKE

Se trata del comando opuesto al comandoGRANT. Se utiliza para retirar privilegios de un usuario. Su sintaxis es muy similar a la sintaxis de GRANT:

```
RSYOD prlv1lagloa ((colua.•I I
OR ele—nto
FROII nomore_u•u•rio
```

Si se han concedido privilegios con la cláusula NITH GRANT OPTIOR, puede revocarlos de la siguiente forma:

```
aavou ALL PRIVILSGU, GIIAIIT
FIIOIIIIOIlbro_uauado
```

Ejemplos de uso de GRANT y REVOKE

Para configurar un administrador, puede escribir:

```
myaql> gr-t all
    -> OD *
```



```
    -> to frecl i<leatified by •...a.123•
    -> witJa gr8&t opti_,
```

Este ejemplo concede todos los privilegios sobre todas las bues de datos a un ~~o llamado Fred con la contra~ amb123 y le permite traspasar dichos privilegies. No le gustará lener un usuano así en su sistema por lo que convienei:evocarlo:

```
myaqb reYOke all pri•Uevaa, ~-t
    -> froa frecl1
```

Seguidamente vamos a configurar un usuario normal sin privilegios.

```
ay,1qb graat "ª89ª
    -> 0ª booka.•
    ·> to aally i<laatified by •..,10123•,
```

Hablamos con Sally para determinar qui desea hacer y le concedemoelos privilegios adecuados:

```
myaql> gr8&t aalaat, iuart, 11.,..ta, ü1ato, ia4ax, altor, anata, drop
    ·> - boob.•
    ·> to aally,
```

Ffjeseen que no necesitamosespecificar la contraseña de Sally pa!'l'realizar esta tarea. Si decidimos que Sally ha intentado realizar algo no apropiado en la base de datos, podemos reducir sus privilegios:

```
ayaql> ra,roka altor, c:r..ta, drop
    ·> OD booka.•
    ·> fr- aally,
```

y posteriormente cuando ya no necesite utilizar la base de datos, podemos révocar todos sus privilegios:

```
myaqh nYOka all
    •> OD booka.•
    ·> f~ aally,
```

Configurar un usuario para la Web

Necesitará configurar un usuario para que sus secuencias de comandos de PHP se conecten a MySQL Podemos volver a aplicar el principio del privilegio mínimo. ¿Quf se permitirá a las secuencias de comandos?

En la mayor parte de los casos, sólo se necesita seleccionar, insertar, eliminar y actualizar filas de tablas. Podemos configurar estas tareas de la siguiente forma:

```
ayaql> graat aoleat, iuort, delata, updato
    ... - booka.•
    ·> to bookor- idutif:1.ed lls, 'llookor-123•,
```

Por razones de seguridad, debería seleccionar una contrasefta mejor.

EL_HOLISTICO@HOTMAIL.COM - 2006

Si utilizamos un semcio de alojamiento Web, por regla general obtendremos acceso al resto de los privilegios de tipo usuario en una base de datos creada para nosotros. Se nos asignad un nombre de usuario y una contrasei\a para el uso de la Unea de comandos (desde donde crear tablas y realizar otras tareas por el estilo) y para realizar las conexiones de secuencia de comandos Web (para consultar la base de datos). Este sistema resulta un poco menos seguro. Puede configurar un usuario con este nivel de privilegios de la siguiente forma:

```
my•ql> vr-t H1—t, i•••rt, up4ate, dalete, iadea, alter, cr—te, c1cop
      •> - JNioka.•
      •> to booko~     id-tiHad     by 'bookor — 12:1•,
```

Proceda a configurar esta segunda versión del usuario ya que es la que utilizaremos en la siguiente sección.

Cerrar la sesión como usuario raíz

Puede cerrar la sesión del monitor de MySQL esaibiendo qui t. Debería volver a registrarse como usuario Web para comprobar si todo funciona correctamente. Si se ha ejecutado la instrucción GRANT pero no puede acceder al intentar iniciar sesión,. puede que no haya eliminado el usuario anónimo durante el proceso de ins~lación. Vuelva a iniciar sesión como usuario raíz y consulte las instrucciones incluidas en el apéndice A sobre cómo eliminar cuentas anónimas. Debería poder iuidax sesi6rt come ll&llari&Web.

Utilizar...la base de.datos correcta

Si ha llegado hasta aquí, debería haber iniciado en una cuenta de MySQL de nivel de usuario lista para probar el ejemplo de código, bien porque acaba de configurMla o porque el administrador del servidor Web se ha encargado de ello.

Lo primero que necesita hacer al iniciar una sesión es especificar qué base de datos desea utiliza. Para ello, puede escribir:

```
my•ql> ••• aambrebd1
```

donde *nombrtbd* es eljombre de la base de datos. Como alternativa, puede evitar el comando use especilicando la base de datos al iniciar la sesión, de la siguiente forina:

En este ejemplo, utilizaremos la base de datos de libros:

```
mymql> ••a~.
```

Al intradbcir este comando, MySQL devolver, una respuesta como la siguiente:

```
Databa•• cllaDgecl
```


Si no selecciona una base de datos antes de empezar a trabajar, MySQL generad un mensaje de error como el siguiente:

```
DJIOII 104' (3DOOOI: II0 Databa•• S.lected
```

Crear tablas de base de datos

El siguiente paso en la configuración de la base de datos consiste en crear las tablas. Para ello puede utilizar el comando CRBATB TABLE de SQL Esta instrucción suele presentar esta forma:

```
CRBATS TAIILl1 ao.wbret•l:ll•  (co1U8DH)
```

Debería sustituir el marcador de posición *nombrrtabla* por el nombre de la tabla que desea crear y el marcador de posición *columnas* por una lista de columnas separadas por comas en la tabla. Cada columna tendr6 un nombre seguido por un tipo de dato.

Volvamos a repasar el esquema de Book-0-Rama:

```
CU.tCIIMlra(CU•to-.rID,   Naae,  Addreaa, City)
Orcser•(Ordllr1D,  CUatonierlD, Amoant, Date)
Booka(ISIIII, Autb.or, Title, Price)
Order_I~...(0:rderID,ISBS,   Qllantity)
.aee11-_a.-·hw•C:tSU.   Pavi-|-
```

El listado 9.1 muestra el código SQL para crear las tablas, asumiendo que ya hayamos qeado la ~ ~e datos lla~~~bo oka. }:nccntrar6 el —igo necesario en el archivo correspondiendo a este capítulo del CD-ROM.

Puede ejecutar un archivo SQL existente, como el cargado desde el CD-ROM a través de MySQL mediante la siguiente secuencia:

```
> ayeql ·la 11.oat ·u bootor._  ·D booka ·P e bookor-.aql
```

(Recuerde sustituir hoat con el nombre de su host.)

El uso de la redirección de archivos resulta bastante pnktico en este caso porque permite editar las secuencias SQL en el editor de texto, deseado antes de ejecutarlas.

Uatado 1.1. bookorllma.aql. Código SQL para crear laá tablas de Book-0-Rama.

```
creat• table caatomer•
( cuatoaerid illt unaigned not null auto_incr...nt pri..ry key,
  —    cll.ar(50) -t   null,
  addreH  cll.ar(1OOI  not IIIIII.
  city cbar(30)  not null
);

create table arder•
( orderid 1-t — .igned —t mall •uto_incr —— t pri-ry ay,
  cuatoaerid ú:at unaigned not .mul.l,
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
   a110UDc e1oac1,.21.
   daca daca noc null
)/

creace cable booke
I  i•bn cbarllll  noc null pri..ry  key,
   aucbor cbarlS01,
   cicle cbarl1ODl.
   price e1oacl4,21
!!

creace cable ordar_ice..
I orderid inc uneigned noc null,
  iabn cbarllll  noc null,
  quancity Cinyinc un•igned,

   pri..ry  key lorderid,  hbn)

I;
creace cable book_reviewe
I
   i•bn cllarllll  noc null pri..ry  key,
   review text
)i
```

Cada tabla se crea con una instrucción CREATE TABLE independiente. Como puede observar hemos creado el esquema con las columnas diseñadas en el último capítulo. Cada columna tiene un tipo de dato listado tras su nombre. Algunas columnas tienen otros especificadores.

Significado del resto de las palabras clave

NOT NULL significa que todas las filas de la tabla dtben tener un valor en este atributo. Si no se especifica, el campo puede dejarse en blanco (NULL).

AtrrO_INCREMENTes una función especial de MySQL que puede utilizar sobre columnas de tipo entero. Significa que si dejamos el campo en blanco al insertar filas en la tabla, MySQL generará automáticamente un valor de identificación exclusivo. Este valor será una unidad mayor que el valor máximo incluido en la columna. Sólo puede tener una columna de este tipo por tabla. Las columnas que especifiquen AtJTO_INCREMENTdeben estar indexadas.

El uso de la palabra clave PRIMARY KEY tras el nombre de una columna determina que la columna se tratará como la clave principal de la tabla. Las entradas de esta columna deben ser exclusivas. MySQL indexará automáticamente esta columna. Ffjese en que en el listado anterior al utilizarlo con customerid en la tabla customera se ha hecho conAOTO INCREMERT.La indexación automática sobre la clave pdncipal se encarga del índice que necesita AOTO_INCRIMERT.

Si especificamos PRIMARY KEY tras el nombre de una columna, sólo se puede utilizar para claves principales formadas por una sola columna. El uso de la cláusula PRIMARY KEY al final de la instrucción order_items es una alternativa. Aquí la

hemos utilizado porque la clave principal de esta tabla se compone de dos columnas. (De esta forma tambifnse crea un índice basado en las dos columnas.)

La palabra clave UJISIGNED utilizada tras un tipo entero significa que sólo puede tener como valor cero o un número positivo.

Tipos de columna

Tomemos la primera tabla como ejemplo

```
creace table cuatoaer•
I cu•toaerid inc unaigned not null auco_incre..nc primary key,
  name char(S0I noc null,
  addre•• cllarl1D0I  noc null,
  cicy char(JDI  noc null
I;
```

Al crear una tabla, deberá tomar una decisión con respecto a los tipos de columnas. En la tabla customers, tenemos cuatro columnas como se especifica en nuestro esquema. La primera, customerid es al clave principal,que hemos especifi~ado directamente. Hemos decidido que sea de tipo entero (tipo int~ y que los ID sean unsigned. También hemos aprovechado la función auto_incr.!D'ent para que MySQL pueda encargarse de ello autOIMticamente y evitamos una preocupación.

Las otras dos columnas contendrán datos de tipo cadena. EnJ!Ste caso hemos seleccionado char como tipo de dato. Este tipo especifica campoiide achura fija a través de corchetes; por ejemplo, name, puede tener hasta *S0* caraeferes.

Este tipo de datos asignará siempre *S0* caracteres de espado de almacenamiento para el nombre, aunque no se utilicen todos. MySQL rellenará los datos con espacios para asignarles el tamaño exacto. Como alternativa se puede utilizar el tipo varchar, que sólo utiliza la cantidad de espacio necesario (más un byte). El tipo varchar absorbe menos espacio pero resulta más lento.

Para los clientes reales con nombres y direcciones reales, las anchuras seleccionadas para estas columnas son demasiado estrechas.

Fljeseen que hemos declarado todas las columnas como MOT NULL. Se trata de una pequeila optimización que puede realizar cuando resulte posible. En un capítulo posterior trataremos el tema de la optimización.

Alguna de las instrucciones CREATE presentan variaciones en cuanto a la sintaxis. Examinemos la tabla ordera:

```
create table ordere
I orderid inc un•igned not null auto_incr...ac primary key,
  cuet—rid int unaigned not null,
  —==c Uoacl&,21.
  daca daca noc null
),
```

La columna amount se especifica como número de coma flotante de tipo float. En la mayor parh! de los tipos de datos de coma flotante, puede especificar la anchura de visualización y el número de decimales. En este caso, la cantidad del

EL_HOLISTICO@HOTMAIL.COM - 2006

?!(lido se establecerá en dólares, por lo que podemos uignar un tamailo total razonablementelargo (una anchura de seis espacios) y dos decimales para los c:m-timos. La columna date lleva asignado el tipo de dato date.

En esta tabla concreta, hemos especificado que las columnas puedan definir la ——d como ROT NOLL. ¿Por qué? Cuando se introduce un pedido en la bue.de daeo., lendremm que ciado en ordera, agregarle lmelementaa a order.;_itw y c:alcular la canticf4d. Puede que no sepamos la cantidad al crear el pedido, por lo que permitimos que sea NULL. La tabla booka presenta algunas c:andedsticas ~ •••

```
cr..te table boOlr•
I  ieba curClll  not null priury key,
   autbor cb8r150!,
   titl• curl1001.
   price floatlt,21
)'
```

En este caso, no necesitamos generar la clave principal porque los ISBN se generan en otra parte. Hemos dejado los otros campos como NOLL porque una libreri& puede conocer el ISBN de un libro antes de conocer su título, autor o precio.

La tabla arder_ i tema muestra cómo crear claves principales multicolwnna:

```
cre~t• table order_it...
I orderid int un•igned not null,
---IIIIIII  cbaz(u+--  ....u..-
  cpaalltity tillyiat un•igned,

  pr:Lmery key iorderid,  iabal
```

Hemos especificado la cantidad de un libro como TINYIRT tJRSIGNED; que equivale a un entero entre 0 y 255.

Como se mencionó anteriormente, las claves principales multicolumna necesitan especificarse con una cláusula de clave principal. &taes la qué se utiliza aqui. Por último, examine la tabla book_reviewa:

```
create table book_reviewa
1
   ielm cbar(13I not null pri•ry key,
   review text
)'
```

Esta tabla utiliza un nuevo tipo de dato, text, que todavía no se ha analizado. Esle tipo se utiliza para texto de gran longitud como un artfc:ulo. Existen algunas variantes de este tipo que se comentarút en una sección posterior.

Para comprender mejor el proceso de creación de tablas vamos a analizar los nombres de las columnas y los identificadores en general, y, a continuación, los tipos de datos que podemos seleccicmar para las colwnnas. Pero en primer lugar, examinaremos la base de datos que hemos creado.

Examinar la base de datos con SHOW y DESCRIBE

Inicie la sesión en el monitor de MySQL y utilice la base de datos books. Para ver las tablas de la bue de datos, escriba

```
myaqb •~  tülu,
```

MySQL mostrad una lista de ~ lu tablas de la base de datos:

```
+-----------------+
| Tablea in booka |
+-----------------+
| book_reviewe
| booka
| cuat-ra
| order_iu•
| ordera
+-----------------+
5 rowa in ••t 10.0,  aecl
```

También puede utilizar la instrucción show para ver una lista de bases de dat.os escribiendo

Si no dispone del privilegio SHOW DATABASES, únicamente ved enumeradas las bases de datos pan las que tenga privilegios.

Puede ver más infonñiaoñ-iobieuñi lalílaconc:ftta, por e}eüq>lo, :ltbrcs, utitizando DESCRIBE:

MySQL mostrar, la informadón suministrada al crear la bue de datos:

```
+--------+------------+------+-----+---------+-------+
| Field  | Type       | IIIII1 | Key | Default | Bxtra |
+--------+------------+------+-----+---------+-------+
  iabn     cbarlll!   |      | PRI |         |
| autbor | cur(SOI    | YES  |     | NIJLL   |
| title  | curl1OOI   | YES  |     | NULL    |
| prica  | floatlt,ZI | YIS  |     | NULL    |
+--------+------------+------+-----+---------+-------+
• rowa in ••t 10.os  aecl
```

Estos comandos resultan útiles para rea>rdar un tipo de columna o navegar por una base de datos aeada por otra persona.

Crear índices

Ya hemos mencionado brevemente los fndic:esya que al diseñar claves principales se crean fndices en dichas columnas.

EL_HOLISTICO@HOTMAIL.COM - 2006

Uno de los problemas IÚS habituales a los que se enfrentan los nuevos usuarios de MySQL es el pobre rendimiento de esta base de datos que creían realmente rápida. Este problema de rendimiento se debe a que no han creado indices en su base de datos. (Se pueden crear tablas sin claves principales ni indices.)

Para empezar, le servirán los índices que ya se han creado automáticamente. Si piensa e;ec:utar varias consultu en una columna que no sea una clave, puede que le interese ai'ladir un índice a la misma para mejorar el rendimiento. Puede hacerlo por medio de la instrucción CREATE INDBX. cuya sintaxis es la siguiente:

```
OU.TE [UIIIOGS:PULLTUTJ IIIDIX nCllllbre_iDClice
OIl IICllllbre_i:üla IIICllllbre_col-_indice [ !loag1tudl J [ASCj,DISC:I, ... 11
```

(Los índices FULLTEXT se utilizan para indexar campos de texto, como veremos en un capitulo posterior.) El campo opcional longitud le permite indicar que sólo los primeros caracteres especificados por dicho campo se indexarán. También puede especificar que un índice sea ascendente (ASC) o descendente (DESC); el valor predeterminado es ASC.

Nota sobre los tipos de tablas

Puede que sepa que MySQLofrece más de un tipo de tabla o motor de almacenami~to. incluyendo algunos tipos compatibles con transacciones. Los analizaremos en un capitulo posterior. En éste, todas las tablas de la base de datos utilizan el m~r de almacenamiento predeterminado: MylSAM.

Identificadores de MySQL

MySQL consta de ~co tipos de identificadores: bases de datos, tablas, columnas e índices (elementos que ya conocemos) y los alias; que examinaremos en el siguiente capitulo. Las bases de datos de MySQL se asignan a directorios en la estructura de archh·os subyacente y las tablas lo hacen a archivos. Este hecho afecta directamente a los nombres que se les puede asignar. También afecta al uso de ':"Byúsculas y Ininúsculas en dichos nombres. Si su sistema o~tivo discrimina entre mayúsculas y minúsculas utilizados en los nombres de los directorios o archivos, los nombres seleccionados para designar las bases de datos y las tablas también lo harán (por ejemplo, en UNIX)o, en caso contrario, no lo harán (por ejemplo, en Windows). Los nombres de columnas y los alias no discriminan entre mayúsculas y minúsculas, pero no puede utilizar versiones diferentes de los nombres en la misma instrucción de SQ~. La ubicación del directorio y los archivos que contengan los datos se establece en la configuración. Puede comprarlo en su sistema utilizando la instrucdón myaqladmin de la siguiente forma:

```
myeqladllin varialll••
```

Estamos buscando la variable datadir.

En la tabla 9.4 se recoge un resumen de los posibles identificadores. La única excepción adicional es que no se puede utilizar ASCil(O),ASCll(255) ni el carácter de comillas en los identificadores (que es muy probable que no utilice).

labia 1.4. Identlflc8dcns de MySQL

Tipo	Longitud máxima	Discrimina entre mayúsculas y minúsculas	Caracteres admitidos
		Según el sistema operativo	tldoeporellll181MOl*mi-vo .,.,. ., nombre de 111'1 dnctoriaeuc:.pc:16ncle'-,, ct1rec..a /, \, y . ;;;?''
Tibia	84	Según el sistema operativo	..-.. = .e voenun,.....deen:lllvo eacepc:16ndelos_... rN/Y.
Columna	64	No	TodollOlClll'8C:le!W
Indice	84	No	Todolla9---
Ala	255	No	Todolla9cel9ClerW

Las reglas son muy abiertas. Desde la versión MySQL 3.23.6, se pueden utilizar palabras reservadas y caracteres especiales de todos los tipos en identificadores, la única limitación es colocar entre comillas inclinadas los caracteres extrat\oe. Por ejemplo:

```
cruce datalla•• ·crear ba•e de dai:o•· 1
```

Las reglas de las versiones de MySQL(anteriores a la 3.23.6) son mAs restrictivas y no lo permiten. Obviamente, conviene aplicar sentido común a toda esta libertad. El hecho de poder llamar a una base de datos 'crear base de datoa· no significa que tengamos que hacerlo. Se aplica el mismo principio aplicado en otros tipos de programación: utilice identificadores descriptivos.

Seleccionar tipos de dato de columna

En MySQLexisten tres tipos búicos de columnas: numérico, de fecha y hora, y de cadena. Cada categoría incluye un gran número de tipos. En este capitulo los resumiremos. En un capitulo posterior se analizarán sus puntos fuertes y sus puntos débiles. Cada uno de los tres tipos dispone de varios tamailos de almacenamiento. Al seleccionar un tipo de columna, el principio general consiste en escoger el tipo más pequeño eit el que encajen los datos. Muchos de los tipos permi~en especi~

EL_HOLISTICO@HOTMAIL.COM - 2006

ficar el tamai\o máximo de visualización al aear la columna. Esta opción se identifica en las siguientes tablas con una M. Si resulta opcional se coloca entre corchetes. El valor máximo que se puede especificar para M es 255. En las siguientes descripciones se utilizan los corchetes para indicar valores opcionales.

Tipos numéricos

Los tipos numiricos son números enteros o de coma flotante. En el caso de los números de coma flotante, puede especificar el número de lugares decimales. En este libro se utiliza D para indicarlo. El valor máximo que se p•ede especificar para Des de 30 o M-2 (es decir, la longitud máxima de visualización menos dos, es decir, un carácter para el punto decimal y uno para la parte entera del número), según cual sea la menor. Para los tipos enteros tambiál puede especificar si desea que sean ONSIGHBD, como se muestra en el listado 9.1. En todos los tipos nummcos, también puede especificar el atributo ZBROFILL. Cuando se muestran los valores de una columna ZBROFILL, se rellenarán con ceros a la derecha. Si especifica una columna como ZBROFILL se le uignará automáticamente el atributo ONSIGNBD.

En la tabla 9.5 se muestran los tipos integrales. En la columna de rangos se muestran los rangos con firma y sin firma en líneas diferentes.

Tabla 9.5. Tipos de datos integrales.

Tipo	Rango	Espacio de almacenamiento (bytes)	Descripción
TIIIYIIIT [IN) J	-l27 •• 12160 •• 255		En18ro•muy l*luellllll
8IT			Sinónimo deTIIIIYllIT
800L			Sinónimo deTIJrYDIT
SNALLlIff [(NI J	-32788••32767 60 .. 65535	2	Enleroll pequello8
NSDIUNilff [IN) J	-83116011••838860760 .•18777215	3	Enleroll de tem8110medio
IIITCINII	·2".2"-160 .. 16777215	4	Entenls~
IIITIGID [(NI J			Sinónimo deINT
BIOIIIT [INI J	-:ZU••2"-1 6 0•.2"' -l	8	Enteros gnilldes

En la tabla 9.6 se recogen los tipos de coma flotante.

Tabla 9.1. Tipoa de datos de coma flotante.

Tipo	Rango	Espacio de almacenamiento (bytes)	Descripción
FLOAT(precisión — t".	Depende de la precisión		SepuedelllmDrs-ni-peclflcar número• de _..._depnicill6n 6nlca o doble.
PLOAT I IN, DI J	t1.175484351 E-38 t3.402823466E+38	4	N6mM, de coma tlotmnte de pl'1ICili6n única. Equ6-,,... a 'PLOAT 141, P9ft> con un ancho de Nuall-aclón yunnúmerode d• clmalN NI)8CIllcado.
DOIIBLB[IN,DI J	t1.7978831348123157E +308 t2.2290731ll85072014E ·30II	8	Númerodecomallolante de pl'1ICili6ndoble. eq... vale a~T 111 peracon un Mdlodevisumllzed6n dado y con un número concretodedecimeML
DCXIBLB PRBCISION (IN, DI I	Como en el cuo anterior		Sin6nlmodeDOOBLB [IN,DIJ.
REALIIN,DIJ	Comoen el caso anterior		Sinónimo de DOU8LB [IN,DIJ.
DBCINAL[IN[,DI 1l	Varia	M+2	Núnwodecoma llatMle almacenadocomochar. El rangodependededeN, la anchura de vtsualimcl6n,
1'1JIIBRIC[IN.DIJ	——— - —— nlllrlor		Sinónimo deDBCIMAL,
DBC[IN,D)J	Comoenetcuo---		SIÍl6lllmO deDSCIMAL.
FIXBD [IN, DI J	Comoenel_.,._		Sinónimo deDBCIMAL.

Tipos de fecha y hora

MySQL admite una gran cantidad de tipos de fecha y hora, como se puede observar en la tabla 9.7. Estos tipos permilen recoger datos en formato numérico o de cadena. Conviene destacar que si se utiliza una columna TIMBSTAMP se establecerá en la fecha y hora correspondiente a la operadón más reciente realizada sobre la fila si no se establece manualmente. Este comportamiento resulta útil para el registro de transacciones.

Tabla 9.7. Tipos de dalDs de fecha Y hora.

Tipo	Rango	Descripción
	1000-01-01 9999-12-31	
	-838:59:59	

EL_HOLISTICO@HOTMAIL.COM - 2006

Tipo	Rango	Descripción
	838:59:511	Tqaena....que .. ....,,_.mucllomn mmpllo de lo que .. pratlÑleque ,-ite.
DATITIM8	100CM>1,0100:00:00 9199-12-3l23:59:59	Una rectiay-"°'9.SelIIIIIIlmillcon .. rarma-to YYYY-NN-DD IIB:NN:SS.
TIMUTMt (CMI J	1170-01.01 00:00:00	UIIII 1119,ca de tiempo, que _.... útil 1*8 te genereció́n • ....._. El form810devi111811za. ció́n depende del v8lar de M (wae le mbla 9.8 lllu9daacanllnuac:l6n).,
	Algúnmomantoan2037	La parte lUP8riDrdal rugo dapende dal Hmite de lu marc•• de tiempo de UNIX.
YUJIIl2l41J	~ (1170-2089) 1901–2155	Un IIIIO.Puacle apac:illc:ar un fannato da 2 • 4 dlgiloa. Cacla_de.._llenaunrangodlferan. ta, como• muNtra.

La tabla 9.8 muestra los diferentes tipos de formatos de visualización de TIMBSTAMP.

Tabla t.l. Tipos de formato de visualización de TIMESTAMP.

Tipo especificado	Formato de visualización
TINBSTNIP	AAAAMMDDHHMMSS
TINBSTANP (14)	AAAAMMDDHHMMSS
TINBSTANP (12)	AAMMDDHHMMSS
TINBSTANP(lO)	AAMMDDHHMM
TINBSTANP(8)	AAAAMMDD
TIMBSTANP (6)	AAMMDD
TIMBSTANP(41	AAMM
TINBSTANP(2)	AA

Tipos de cadena

Los tipos de cadena se clasifican en tres grupos. En primer lugar, están las antiguas cadenas piulas, es decir, pequeftos fragmentos de texto. Se trata de los tipos CRAR (cancter de longitud fija) y VARCHAR (cancter de longitud variable). Puede hpecificar la anchura de cada una. Las columnas de tipo CHAR se rellenarán con espacios hasta cubrir la anchura múima, independientemente del tamai\o de los datos. (MySQL elimina los espacios que sobran a la derecha de los tipos CHAR al recuperarlos y de los tipos VARCHAR al almacenarlos.) En un capítulo posterior

examinaremos las ventajas y desventajas en tirminos de espacio y velocidad de estos dos tipos.

En segundo lugar, están los tipos TEXT y BLOB. Estos tipos disponen de varios tamaños. Equivalen a texto largo y datos binarios, respectivamente. Los BLOB son objetos largos binarios. Pueden incluir todo lo que se necesite, por ejemplo, imigenes o sonidos.

En la práctica, las columnas BLOB y TBXT son iguales con la salvedad de que TEXT discrimina entre mayúsculas y minúsculas, y BLOB no lo hace. Como estos tipos de columna pueden contener grandes cantidades de datoe, es necesario tener en cuenta una serie de consideraciones, como se veril en un capítulo posterior.

El tercer grupo consta de dos tipos especiales, SET y UOM. El tipo SET se utiliza para especificar que los valores de esta columna deben proceder de un conjunto de valores dados. Los valores de columna pueden contener más de un valor del conjunto. Cada conjunto especificado puede contener un múimo de 64 elementos.

EIltJM es una enumeración. Es muy similar a SET, con la excepción de que las columnas de este tipo sólo pueden tener uno de los valores especificados o NULL y el máximo número de elementos de la enumeración serA de 65.535.

En las tablas9.9, 9.10 y9.11 se recogen los tiposdedatosdecad~. J,.a tabla9.9 muestra los tipos de cadena comunes.

Tabla t.t. Tipos de cadena habituales.

Tipo	Rango	Descripción
[IJIATIOIIAL) CIIAII (M) (BIDIIY | ASCII | IINICODIJ	pe 0 a 255 ~	Cadenadelangllud ljaM, clondeNN .... blaoa anlnt 0 y 255. La ..,_,. cllift IIATIOIIALNpecib que 18 debe ..-zar el conjunlodeCll'aClerNpqdetWmlnaclo. Se.... dlllllpo~ ..-..... MySQl. pero .. lnc:luye ya q.. forma parte del _.....,, ANSI SQL L• p9labr8 ~ BIXAJLYNpeCiftcaqualaadatoadeberfan llwa•-linoclac:r'.mllwantrema-,,:acula ynniac:ulaa. (O.manaraprede-lllmlkl8da. dlacrtmina entre mayúac:ulaa y mklúac:ula.) L8palabra Claft ASCII .... pacillc:8 qua ..... cokNmll N ulilIz.a el mnjunla de carac:18realIIIIn1. La palabra ctevaUIIICODl:lndk:8quaNutllIu.icon-junlode----UCI.
CIIAIt		SIn6nimodeaWt 111
[IIATIOIfAL) VAJLCIIARINI (BIJWIY)	De 1 a 255 csrKI8rN	lgualque•anllrior. conte~que • longllud.. "8llabllt.

La tabla 9.10 describe los tipos TEXT y BLOB. La longitud de un campo TEXT en caracteres es el tamafto máximo en bytes de los archivos que podÑft almacenarse en dicho campo.

EL_HOLISTICO@HOTMAIL.COM - 2006

. Tmla 1.10. Tipos TEXTy Bl08.

T111YBLOII	2" ·1 <• decir, 255)	Un mmpoar.oa peque11o
T11ffTBXT	2" ·1 (H decir, 255)	UnQfflpOTUTpequello
ar.oa	211 -1 ‹• decir, 85.535)	Unc:ampoai.oa de ta111111o norm.,
TUT	211·1 (•decir, 85.535)	UnmmpoTUTdei.m.llonarm•
IIBDIUIIILOI	2'"·1 (Nclecir, 18.777.215)	UnQl!lpOIUIBde.....noilunnedio
IIBDIUJITUT	2,.•1 (Hclecir, 18.777.215)	UnCMIPOTUTde 181111111ointennedio
LOIIOIu;JI	211·1 (Ndllcir, 4.294.967.295)	UnC9111POILOBdeiarn.llogrande
LOIIOTDT	211.1 (Nclecir, 4.294.967.295)	Un QIIIIIOflXTdemmellogr9~

La tabla 9.11 muestra los tipos ENUM y SET.

Tabla 1.11. ~ SET y ENUM.

ENUM('valor1','valor2',...)	65.535	LMeoi.-1tideNIB!lpol6k! p,eedeH 9 IIULL.
SIT I'valorl ' ,•valor2 ', ". I · 114- -		L•columnudellletiPopuNencon-i.wwtconjunlode,,.....NpeCllIce-dos OIIULL.

Lecturas adicionales

Si daea obtener más información, puede consultar el manual en lfnea de MySQL para configurar una base de datos en ht,ep://www.mysql.com.

A continuación

Ahora que ha aprendido a cror usuarios, bues de datos y tablu, puede concen-lrUN en intenctuar con la bue de datos. En el siguiente capítulo, veremos cómo intloducir datas en tablu, cómo actualizarlos y eliminarlos, y cómo consultar la buededalm.

EL_HOLISTICO@HOTMAIL.COM - 2006

10

Trabajar con la base de datos de MySQl

En este capítulo vamos a analizar el lenguaje de consulta estructurado (SQL) y su uso para consultar bases de datos. Continuaremos con la base de datos Book-0-Rama. En concreto, veremos cómo añadir, eliminar y actualizar datos y cómo realizar preguntas a la base de datos.

En este capítulo se abordarán los siguientes aspectos:

- Concepto de SQL
- Añadir datos a la base de datos
- Recuperar datos de la base de datos
- Combinar tablas
- Utilizar subconsultas
- Actualizar registros de la base de datos
- Modificar tablas tras su creación
- Eliminar registros de la base de datos
- Eliminar tablas

Comenzaremos por explicar el significado de SQL y por qué resulta útil entender el concepto en el que se basa este lenguaje. Si no ha creado la base de datos de Book-0-Rama, necesitará hacerlo para poder ejecutar las consultas SQL de este capítulo. En un capítulo anterior encontrará las instrucciones.

EL_HOLISTICO@HOTMAIL.COM - 2006

Concepto de SQL

SQL equivale a Lenguaje de consulta estructurado. Se trata del lenguaje estándar para acceder a los sistemas de administración de base de datos (ROBMS). SQL se utiliza para almacenar y consultar datos desde y hasta una base de datos. Se utiliza en sistemas de base de datos como MySQL. Oracle, PostgreSQL, Sybase y Microsoft SQL Server entre otros.

Existe un estándar ANSI de SQL, y los sistemas de bases de datos como MySQL suelen implementarlo. Sin embargo, existen diferencias sutiles. entre el SQL estándar y el SQL de MySQL. que en algunos casos está previsto integrar en el estándar y en otros resultan deliberadas. A medida que avancemos, iremos indicando los casos más importantes. En el manual en linea de MySQL se recoge una lista completa de las diferencias entre el SQL de MySQL y el SQL ANSI de cada versión. Esta página se puede encontrar en el siguiente URL así como en otras muchas direcciones:

```
bttp://,,,,.,.ay•ql.C011/dóc/•n/Coapetlblllty.btal
```

Puede que haya oído hablar de los lenguajes de definición de datos (DDL), utilizados para definir bases de datos, y de los lenguajes de manipulación de datos (DML), utilizados para consultar bases de datos. SQL trata estos tipos de bases. En U'1 capítulo anterior, vimos la definición de datos (DDL) en SQL; por lo que ya la hemos utilizado. Este lenguaje se utiliza al configurar inicialmente una base de datos. Utilizanlllos losaspecto& DML de SQL con mucha mayor frecuencia porque nos servirán para almacenar y recuperar datos reales de una base de datos.

Añadir datos a la base de datos

Para poder trabajar con la base de datos, es necesario almacenar datos en ella. La forma más habitual de llevar a cabo esta operación consiste en utilizar la instruc·á6n IBSIRT de SQL Recuerde que los RDBMS contienen tablas, que a su vez contienen filas de datos organizados en columnas. Cada fila de una tabla suele describir un objeto o relación del mundo real y los valores de las columnas para dichas filas almacenan información sobre el mundo real Podemos utilizar la instrucción IRSIRT para colocar filas de datos dentro de la base de datos.

A continuación, se recoge la sintaxis habitual de una instrucción INSBRT:

```
t•sDT    (IIÍTOI túl•    ( (col111111•J, C:OlllmMIZ, col-..J,  ...  11  VALUBS
(valorl,    vüorZ,    v•lorJ,  ...  JI
```

Por-ejemplo, para insertar un registro dentro de la tabla Cuatomera de Book-Q. Runa, puedeescribir

```
l1111eñ lntO" cu•t — r•  ••lo•
   (JIULL,  •.JUll•  Salth",  •2s  Oak  Street•,   •IJ.rport We•t ª11
```


Como puede observar, hemos sustituido *tobbr* por el nombre de la tabla en la que queremos colocar los datos y los w/orrs con valores espedficos. Los valores de este ejemplo se encierran entre comillas dobles. En MySQL, las cadenas deberían encerrarse siempre entre comillas simples o dobles. (En este libro utilizaremos ambu.) Los números y las fechas no necesitan comillas. Existen ciertas cosas interesantes en los que fijarse en relación a la instrucción INSBRT.

Los valores que especificamos se utilizarán para rellenar las columnas de la tabla en orden. Si desea rellenar sólo algunas columnas o si quiere especificarlas en un orden diferente, puede enumerar las deseadas en la parte reservada para las columnas de la instrucción. Por ejemplo:

```
ln•ert lnto cu8t011er• (naae,  cltyl valo•
("M•li••• Jone••,  "Mar Mar Goon Nortb"l!
```

Este enfoque resulta útil si sólo se dispone de datos parciales sobre un registro dado o si algunos campos del registro son opcionales. Puede obtener el mismo efecto con la siguiente sintaxis:

```
in•ert into cu•toa.r•
•et IUÍ.. •"Mlcbael Arcber•.
   •ddr••••"12  Adderley Aveno•,
   cicy••Leeton•;
```

Como puede observar, especificamos un valor NULI para la columna c:uatomerid al agregar a Julie Smith e ignoramos dicha columna al agregar el resto de los clientes. Puede que recuerde que al configurar la base de datos, creamos la columna custo|||erid como clave principal de la tabla customers por lo que puede que le resulte extrai\o. Sin embargo, especificamos el campo como AtJTO_INCRIMINT, lo que significa que si insertamos una fila con un valor NOLLo sin ningún valor en este campo, MySQL generará el siguiente número en la secuencia de incremento automática y la insertará sin nuestra intervención, lo cual resulta bastante útil.

Tambiin puede insertar varias filas dentro de una tabla de una sola vez. Cada fila debería incluirse en su propio conjunto de corchetes y cada conjunto de corchetes deberia separarse por medio de una coma.

Sólo se permiten algunas variantes con IRSBRT. Tras la palabra INSBRT, puede añadir LOW_PRIORITY o DBLAYBD. La palabra clave LOlf_PRIORITY indica que el sistema debe esperar y tener que realizar la operadón posteriormente cuando no se lean datos de la tabla. La palabra clave DILAYBD significa que los datos aftadidos se almacenarán en búfer. Si el servidor está ocupado, se pueden seguir ejecutando consultu sin tener que esperar a que termine la operadón INSBRT.

Tras ello, también puede especificar IGNORI, lo que significa que si intenta añadir filas que generen una clave-principal duplicada. éstas se ignorarán. Otra alternativa consiste en especificar ON DUPLICATIICBY OPDATI expreai6n al final de la instrucción INSIRT. Lo puede utilizar para cambiar el valor ~uplicado por medio de una instrucción UPDATB normal (que analizaremos más adelante).

Hemos reunido algunos datos de ejemplo para rellenar la base de da~. Se trata de una serie de simples instrucciones IRSDT que utilizan el enfoque de inserción

EL_HOLISTICO@HOTMAIL.COM - 2006

multifila comentado. La sec:uentja de coa,andos que realiza esta tarea podn encontrarla en el CD que se adjunta al libro, en el archivo correspondiente a esta carpeta. Tambi4n se recoge en el listado 10.1.

uatac1o· 10.1. book_lnNrt.lql. SQL para completarlas ..... de Book-0-Rama.

```
in•ert into c:uatomer• valuee
   13, •Julie S•itb•, •2s oak Street•, •Airport W.•t"I,
   14, •Alua Wong", °1/47 Baine• Avenue•, •a0X a111•1,
   IS, °Nicbelle Artbur•, •357 Sortb load•, "Yarraville•i.,

in•ert into order• value•
   IIIULL, S, 19.91, o2000·04·02"),
   IIIULL, 3, 49.99, •2000-04-15•1,
   IIIULL, 4, 74,91, o2000·04·19"1,
   IIIULL, 5, 24. H, o2000·05·01"1 ,

in•ert into boolce value•
   (•0-172-31197·1", "Micbael No~an•, •Java 2 tor Profea•ional
   1"0·172·31745·1", °Thoma• Down", °Installing Debian GIIU/Linux•, 24.991,
   Developer••, 34,991,
   1"0-672-31509·2", "Pruitt, et al.•, •Teaeb Your•elt OINP in 24 ROUZ'a•,
   24.Hi,
   ("0·'72-317'9·9", "Tbomaa ScbeDk•, •caldera OpenLiDUa Syat- Adiú.ni•tratton
   Dnleaabed•, 49,9911

insart into order_tc-. valuee
   (1, •0-,,2-31,,1-1•. 21,
   12. •0-,12-3111,-, •• 11.
   (1, •0-,,2.3111,-•~ .11,-
   IJ, •0-112-3150,-2•. 11.
   (4, •0-'72-31745·1", 311

ineert into boolc reviewe value•
   ("0·172·31697:1", °Norgan•e book ta elearly written and goe• vell beyaad
        -•t of tbe baeic Ja- boolc8 out tbere.•>,
```

~e ejecutar esta secuencia de coa,andos a trav& de MySQL de la siguiente forma:

```
>mysql -h host -u bookorama -p < book_insert.sql
```

Recuperar datos de la base de datos

La !!'lúla da carga de SQt. es la instrucción SBLBCT. Se utiliza para recuperar datos ele una base de data. selecc:ionando las filas que coináden con los criterios especificadOIde una tabla. La instrucción SBLBCT consta de una gran cantidad de opciones y formas de uso.

EL_HOLISTICO@HOTMAIL.COM - 2006

La forma búic:a de una instruc:ción SBLBCT es la siguiente:

```
SSLIICT |opcioae•l elemento•
[urro detallee_areJa1vol
FION tablu
I llllDS eolld1e1oaeel
[ OIIGIIP 8Y Upo_grvpo I
I IIIWIIGdefln1c16a de ddnde I
| ,()RI)ft IY t1po_ozden J
I LINIT cr1ter>o•_ll•ite 1 ,
[PIIOCIIDOIIS 11«-bre_J roeed (a ——— to•I I
[opc1oaee_bloqueol
```

Comentaremos cada una de estas c1'usulas de la instraa:ión. En primer lugar, vamos a examinar una consulta sin ningún pal'llmetro opcional que selecciona algunos elementos de una tabla dada. Por regla general, se trata de columnas de la tabla. (Tambifn pueden ser los resultados de cualquier expnsión de MySQL Analizaremos algunos de los m6s útiles en una sec:ci6n posterior.) La siguiente consulta enumera los contenidos de la columna name y ci ty de la tabla cuatomera:

```
•elect naM, city
fr- cu•t—ra,
```

Esta COIIIUlta devuelve el siguiente resultado, siempre *y* mando haya introduádo los datos de ejemplo del listado 10.1 *y* las dos instruc:cioms DISBRT de ejemplo:

-	city
Julie S•ith	Airport IIHt
Alan llong	9ax BU 1
Nicbslle Artbur	Yarraville
NelhN .roa..	hr Nar Ooon Nortb
Nicbsel Arcber	Lsaton

Como puede observar, tenemos una tabla que contiene los elementos selecáonados (name *y* city) de la tabla especificada, customer•. Los datos 1e inuestrm para todas las filas de la tabla cuna.era. Puede especific:artantas columnas como desee de una tabla enumermdolas tras la palabra clave SILBC'I'. TambWn puede especificar algunos otros ele.._tos. Uno útil es el operador c:omodfn, •, que selecciona todas lu columna de la tabla o tablas especificadas. Por e;tmplo, para recuperar todas las columnas *y* todas las filas de la tabla order_iteu, utilizaríamos

```
•elect •
f~ order_it-•1
```

que devolver, el siguiente resultado:

orderid	iebn	f quaa~il:y f

```
+---------+---+----------+----------+
        1 | o-,72•31,97·1 |        2 |
        2 | 0-,12-311,9.9 |        1 |
        3 | 0-,12-317,9•9 |        1 |
        3 | 0-,12-31509-2 |        1 |
        4 | 0-,12-31745·1 |        3 |
+---------+---------------+----------+
```

Recuperar datos con criterios específicos

Para poder acceder a un subconjunto de filas de una tabla, necesitamos especificar algunos criterios de selección. Para ello se puede utiliza!' la d6usula WHERE. Por ejemplo,

```
aelecc •
from ordera
vhere cuacCl1lerid • S,
```

seleccionar4 todas las columnas de la tabla de pedidos pero sólo las filas con el customerid 5. El resultado seria el siguiente:

```
+---------+-----------+--------+-----------+
| orderid | cuaCCl11erid | amounc | dace
+---------+-----------+--------+-----------+
|. 1                  S |  H.98  | 2000·04·02 |
| 4                   S |  24.99 | 2000•05·01 |
+---------+-----------+--------+-----------+
```

La cl4usula WHBRE especifica los criterios utilizados para seleccionar filas concretas. En este ca,o, hemos seleccionado las filas con el Id. de cliente s. El signo igual se utiliza para probar la igualdad (tenga en cuenta que es diferente con:~ pecto a PHP y que resulta sencillo confundirlos cuando se utilizan juntos).

Ademú efe,la igualdad, MySQL admite un conjunto completo de operadores de comparación y de expresiones regulares. En la tabla 10.1 se recogen las más comunes. Tenga en cuenta que no se trata de una lista completa (si necesita algún operador no enumerado, consulte el manual de MySQL).

Tabla 10.1. Operadoras de comparación Otiles para las c:léusulas WHERE.

	.....	cuacoiurid • 3	Comp,uelle al doe Y8lorN —'j''''
"	W.,O,que	i!lpOrCe > 60. 00	Comp,uelle II un Y8lor • ~queolnl
⊂	Menor que	imporce e 60.00	Comp,uelle II un v8lor • —— queoáa
"•	Mayoro lgual que	imporce >• ,o.oo	Comprueb9 II un vmlores ....yoro lgual que otro
<.	Menor o lgmlque	iaporce e• ,o.oo	Compruebe II un vmlilr es menor o lgual que o1lo
1•OC>	Dfflnnte	cancidad 1• o	Compruab• II doa 'lllklnla no–lgu••
IS 1'0T IIOLL	nld	dirección ia noc null	Cclmpruebellelc.mpocon-llene un valor
IS RULL	ntd	dirección ea null	Compruebm al el c:ampo no — ⁿ* — .Un'llllol'
BITIIID	ntd	imporce becveen	Comp,ueba II un vmlol' N mayor o igwl que un Vlllol' mlnlmoo menor o lgual que unv81orlMldmo
IR	ntd	ciudad in (•Carlcon•, •Noe • I	Compiueba II un valoree Incluyedentro de un Cllfljun-lod9do
1'0T IR	ntd	ciudad in 1•carlcon•. •Moe•J	Compruebllllun'llllornoee incluye en un-..-
LID	Comlspondenm depellÓII	noebre like (•Pred t•)	Comprvllbnfun~ c:lde c:on un P11trón utHIDn-do lllmplN pmtronA de c:o-miapondenc:illdeSQL
1'0T LID	Correspondencia de patrones	nombre noc like ("Pred t•t	Compruebli al un valor no c:olnc:ld•c:onunp.arón
RBGBXP	Expresión regular	nombre regexp	Compruabaal unv81orc:oln-c:lde c:on 111111 exprealón reguter

Las últimas tres entradas de la tabla hacen referencia a LI ICE y REGEXP. Se trata de operadores para la búsqueda de correspondencias en forma de patrón.

LIICE utiliza correspondencias de patrón simples de SQL Los patrone1i"pueden consistir en texto normal mú el caraicter 1 (porcentaje)para indicar una coincidencia con cualquier número de caracteres y el cancter_ (guión bajo) para sustituir a un solo car6cter en la búsqueda de correspondencias.

La palabra clave REGEXP se utiliza para las b11squedas de correspondencias con expresiones regulares. MySQLutiliza expresiones regulares de POSIX.En lugar ~e REGEXP, puede utilizar RLIKE ya que son equivalentes. Las expresiones regulares de POSJX tambim se utilizan en PHP. Puede leer mú al respecto en un capítulo anterior.

Puede probar una gran cantidad de criterios de esta forma y combinar los operadores con ARO y OR. Por ejemplo:

EL_HOLISTICO@HOTMAIL.COM - 2006

```
Nlec:C •
fr- orden
wbare cuaC011erid • l or cuacomitrid • •,
```

Recuperar datos desde varias tablas

A menudo, para responder a una pregunta desde la base de datos, es necesario utilizar datos incluidos en más de una tabla. Por ejemplo, si quiere saber los clientes que realizaron pedidos durante el presente mes, necesitará buscar en la tabla customers y en la tabla orders. Si además quiere saber qué.artículos pidieron, necesitará consultar la tabla order items.

Estos datos se encuentran distribuidos en varias tablas porque hacen referencia a objetos ~dependientes del mundo real. Éste es uno de los principios del buen disei\o de bases de datos comentado en un capítulo anterior.

Para reunir esta información en SQL, debe aplicar una operación denominada combinación. Esta operación consiste en unir dos o más tablas para seguir la relación entre los datos. Por ejemplo, si deseamos ver los pedidos realizados por Julie Smith, necesitaremos examinar la tabla customers para buscar el identificador de Julie y, a continuación, mirar en la tabla orders para buscar los pedidos que tengan dicho identificador. Aunque las combinaciones resultan conceptualmente sencillas, son una de las partes más sutiles y complejas de SQL. En MySQL se impiementan varios tipos diferentes de combinación y cada uno de ellos se utiliza para una finalidad diferente.

Combinaciones sencillas de dos tablas

Comenzaremos por examinar algunas instancias de SQL para consultar la información sobre Julie Smith de la que acabamos de hablar:

```
selecc orders.orderid, orders.amounc, orders.dace
fr- cuaca.ers, orders
wbere cuacomers.nama • •Julia Smicb•
and cuac011era.cuaccmerid • ordars.c:uacomerid;
```

El resultado de esta consulta es

orderid	.-une	date
2	.,.,,	2000-04·15

Tenemos que fijamos en varios elementos. En primer lugar, como es necesario obtener información de las dos tablas para contestar a esta consulta, se han enumerado ambas.

También se ha especificado el tipo de combinación, posiblemente sin ser consciente de ello. La coma situada entre los nombres de las tablas equivale a escribir

INHBR JOIII o CROSS JOIN. Éste es un tipo de combinación al que a veces también 54. ~ce ~erencia como combinación completa o el producto cartesiano de tablas. S1grufi~ toma las tablas indicadas y crea otra con ellas; esta tabla debería constar de una fila para cada combinación posible de filas de cada una de las tablas indicadas, ya tenga sentido o no".

En otras palabras, obtenemos otra tabla, que incluye todas las filas de la tabla customers que se correspondan con cada fila de la tabla orders, independientemente de si un cliente realizó un pedido concreto.

En muchas ocasiones el resultado no tiene mucho sentido. Por regla general, lo que queremos es ver las filas coincidentes, es decir, los pedidos realizados por un cliente dado y la información correspondiente del cliente.

Para lograrlo se utiliza una condición de combinación en la cláusula WHBRB. Se trata de un ~po especial de instrucción condicional que explica qué atributos muestran la relaaón entre dos tablas. En este caso, la condición es la siguiente:

```
cuscomers.cuscomerid • orders.cuscomerid
```

que indica a MySQL que ponga filas en la tabla de resultados si el valor customerid de la tabla customers coincide con el valor custo1Mrid de la tabla orders.

Al agregar esta condición de combinación en la consulta, hemoshdo otro tipo de combinación llamada combinación de igualdad.

La nota~ón de punto utilizada deja claro a qué tabla pertenece una columna dada, es dear,customers. customerid hace referencia a la colwnM customerid de la tabla customers y order. customerid hace referencia a la columna customeridde la tabla orders.

Estanotación es obligatoria si el nombre de una columna resulta ambiguo es decir si aparece en mú de una tabla. También se puede utilizar para distinguir los'nomb~ de co~wnnas utilizados en diferentes bases de datos. En este ejemplo, se ha utilizado la notación tabla. columna. Puede especificar la base de datos con una notación basededatos. tabla. columna, por ejemplo, para probar una condición como

```
booka.orders.cuata.erid • ocber_db.orders.cuat-.rid
```

Sin embargo, puede utilizar la notación de punto para todas las referencias de columna de una consulta. Esta opción es una buena idea, en especial cuando las consultas comienzan a complicarse. Su uso l\o es obligatorio en MySQL pero las cons~ltas resul~ mucho más legibl!!Sy sendllas de mantener. Como puede apreaar, ~ utilizado esta convención en el resto de la consulta anterior, por e,emplo, al utilizar la condición

```
cuat-.r•.name • 'Julie Smitb'
```

La columna na- sólo aparece en la tabla customers por lo que no necesitaremos especificarlo. MySQL no se equivocará. Para los humanos, sin embargo el nombre sin mú resulta vago, por lo que el significado de la consulta resultani ~ claro si se especifica tomo customer. na-.

EL_HOLISTICO@HOTMAIL.COM - 2006

Combinar varias tablas

Cuando se combinan más de dos tablas la operación se complica. Como regla general, en las condiciones de combinación se combinan las tablas de dos en dos. Es como seguir las relaciones enne los datos de una tabla a otra tabla a otra tabla, etc.

Por ejemplo, si queremos saber qué clientes han pedido libros sobre Java (por ejemplo, para poder enviarles información sobre un nuevo libro), necesitaremos rastrear dichas relaciones a través de bastantes tablas.

Tendremos que buscar clientes que hayan realizado al menos un pedido que incluya un libro de Java. Para acceder a la tabla orders desde la tabla customers podemos utilizar la columna customer id como hicimos anteriormente. Para acceder a la tabla order_items desde la tabla orders, podemos utilizar la columna orderid. Para acceder a un libro especifico de la tabla books desde la tabla order_items, podemos utilizar el ISBN. Tras establecer estos enlaces, podemos buscar libros que incluyan el tirmino Java en el titulo *y* recuperar los nombres de los clientes que hayan comprado alguno de ellos.

Examinemos la consulta que realiza estas operaciones.

```
••l•ct cu•t0tler•.11a11e
frot1 cuecamere, ordera, order_it ..., boolca
v11are cuatoiaera.cu•to-.rid • order•.cu•to-.rid
and ordera.orderid • order_it....orderid
and order it.... i•bn • book•.i•bn
IUICI booka:title 11ke ',Java,•¡
```

Esta consulta devuelve el siguiente resultado:

```
|.na~                L
|------------------
| Michell• Arthur |
|------------------
```

Fíjese en que seguimos los datos a través de cuatro tablas diferentes *y,* para hacerlo con una combinación de igualdad, necesitamos tres condiciones de combinad6ri diferentes. Por regla general, se necesita una condición de combinación para cada par de tablas que se desee combinar, por lo que el total de condiciones de combinación será igual al número de tablas que se desean combinar menos uno. Esta regla puede resultar útil para consultas de depuración que no funcionen. Compruebe las condiciones de combinación *y* asegúrese de haber seguido la ruta desde lo que sabe a lo que desea saber.

Buscar filas que no coincidan

El segundo tipo principal de consultas que utilizaremos en MySQL son las combimciaw por la izquierda. En los ejemplos anteriores, sólo se han incluido las filas ~- en~ las tablas. En ocasiones, se necesita recuperar las filas que no c~ Por e,emplo, los clientes que no hayan realizado nunca un pedido o los libros que no se hayan pedido nunca•

La f0'"!8 ~ ~lla ~e ~pondera este tipo de cuestión en MySQL es utilizar una combmac16n por la JZquierda. En estas combinaciones se comparan fi1u a partir de ~ condición de combinación dada enne dos tablas. Si no existen filas coincidentes en la tabla de la deiecha, se agregari una fila al resultado que contenga valores NOLL en las columnas de la i:lerecha.

Veamos un ejemplo:

```
••lect cuatomera.cuata..rid, cu.to-.ra.na ... ordera.orderid
fra11 cuat011er• left join order•
on cu•tomer•.cuatomerid • ordera.cu.to-.rid¡
```

La consulta SQL utiliza una combinación por la izquierda para combinar customers con ordera. La combinación por la izquierda utiliza una sintaxis un tanto diferente en la c~didón de combinación. En este caso, dicha combinación va en una cláusula ON especial de la instrucción SQL

El resultado de la consulta es el siguiente:

cu•t011erid	n..e	orderid
1	MeliHa JonH	NULL
2	Micbael Arcllar	IIULL
3	Julia S•ltll	2
4	Alan NoDg	3
5	Micllalle Arthur	1
5	Micllalle Artllur	4

Este resultado muestra que no existen Id. de pedido coincidentes para Melissa Jones y Micbael...Archer.porque dichos Id, son-tltJI;t,.—

Si sólo deseamos ver los clientes que no hayan realizado ningún pedido, podemos recuperarlos utilizando dichos valores NULL en el campo de clave principal de la tabla de la derecha (en este caso orderid).

```
aelect cu•toaaera.cuat011erid, cuatomera.naaie
froe cu•toiaer• left join ordera
uai119 |cuatoeerid|
v1lera ordera.orderld i• null;
```

El resultado ser'=

cut—rid	-
1	MeliHa JODH
2	Micbael Arcllar

Como observad, hemos utilizado una sintaxis diferente para la co~ldón de combinación en este ejemplo. Las combinaciones por la izquierda admiten la sintaxis ON utilizada en~ primer ejemplo o la sintaxis t1SING del segundo. Tenga en cuenta que la sintaxis OSING no especifica la tabla de la que procede el atributo de

EL_HOLISTICO@HOTMAIL.COM - 2006

combinación. Por esta razón, las columnas de Ju tablas deben tener el mismo nombre si queremos utilizar USIHO.

Utilizar otros nombres para designar tablas: los alias

A menudo resulta pr4ctico, y a veces necesario, poder utilizar otros nombres para hacer referencia a las tablas. Es lo que se conoce como alias. Los alias se pueden crear al principio de una columna y utilizarlos a lo largo de todas ellas. Suelen resultar prácticos por cuestiones de brevedad. Por ejemplo, examine la consulta que escribimos anteriormente pero escrita con alias:

```
••l•et e.na•
troa eu•tOMr• •• e, ordera •• o, order_it... •• oi, !looka •• b
wbar• e.cuatOMrid • o.cuatOMrid
aad o.orderid • oi.orderid
aad oi.i•bn  • b.iabn
and b.title  like •1Javal'1
```

Al declarar las tablas que vamos a utilizar, agregamos una djusula AS para establecer el alias asociado a dicha tabla. Tambim podemos utilizar alias para columnas. Volveremos sobre este tema al analizar las funciones de agregación.

El uso de alias es obligatorio cuando se trata de combinar una tabla consigo misma. Suena mú complicado y extraño de lo que parece. Resulta útil, por ejemplo, 'si queremos buscar filas de la misma tabla con valores comunes. Si queremos buscar clientes que vivan en la misma ciudad (por ejemplo, para establecer un sitio de lectura) podemos asignar a la tabla (customers) dos alias düerentes.

```
••leet el.D&M,   c:2.name, el.eity
froa euat011era •• el, cuatOMra••  e2-
wbare el.eity  • e2.eity
aad el.na•  I• c2.na•,
```

Lo que estamos haciendo búicamente es simular la existencia de dos tablas customera diferentes, el y ea, y realizar la combinación sobre la columna city. Como observm tambiét necesitamos la segunda condición, cl .naa. I• ea .name, para no recuperar todos los clientes por tener el mismo nombre.

Resumen de los tipos de combinación

En la tabla 10.2 se recogen los diferentes tipos de combinaciones vistos. Existen otros, pero los incluidoe en la tabla son los más importantes para nuestros propósitos.

Tablll 10..2. Tipos de c:ombinacl6n en MySQL

Producto cartesiano	Se comblmn todas lu fila de todas la tabla. Se ulllza lnc:luJendo una coma entre los nombl'9s de la tablas y sin una cléusula IIBBRB.

Comblnacl6n cruzada	Igual que tas anteriores • T~ se puede utlfzar especlflc:endo... palabrU clave CROSS JOIR enn to. nombres de la tablas combinadas.
Combinación interna	Sen,jntlc:emente. resulta equivalente a ta coma. Tambi6n se puede eapeclflcar utlllzando tas palabras clave INRD JOIN. Sin Uria condk:i6n IIHBU rasulta equlv• lente ~ una combinación total. Por regla general, se espec:iflca una condlcl6n WRBU para obtener una auttntlca combinaci6n lnterna.
Comblnac16n de Igualdad	Udllza una expresl6n condlc:lonal con un signo • para buscar fllu coinddentes entre tas dlfer9ntn tablas de la combinac:i6n. En SQI, equivale a una combinación con una déusula IIHBRB.
Combinación por la izquierda	Busca filas coincidentes entre tablas y 1'9hna las fllá no coincidentn con valores NULL. En '.&QL se utlllza con tas palabras clave LBPT JOilf. Se uWlza para buscar valores que falten. También se paede utilizar la combinación RIGHT JOilf de menara equivalente.

Recuperar datos con un orden dado

Si d~ mostrar en un orden concreto las filas recuperadas en una co.;.~lta, puede utilizar la cl¸usula ORDER BY de la instrucción SELECT. Esta funciori resulta pr4ctica para presentar los resultados en un formato legible para humanos.

La djusula ORDBR BY se utiliza para ordenar las filas de una o varias columnas listadas en la cláusula SELECT. Por ejemplo,

```
••lect naM, addr•••
froacuetOMre
order by na_,
```

Esta consulta devuelve los nombres y las direcciones en orden alfabético por el nombre:

na-	adclrHa
Alan Wong	1/47 Rain•• Avenue
Juli• Saith	25 Oek Street
Meli••• Jonea	
Nie.bael Arcber	12 Adderley Avenue
Niehelle Artbur	357 lforth Roed

EL_HOLISTICO@HOTMAIL.COM - 2006

Al utilizar el formato de nombre y apellido, los registros se ordenan por el nombre. Si desea ordenarlos por el apellido, necesitará utilizar dos campos diferentes. El orden predeterminado es ascendente (de la a a la z o numéricamente hada arriba). Puede especificar este orden mediante la palabra clave ASC:

```
••lect IIIU19,  addre••
fr-  cu•t-r•
order by -    ••cl
```

Tambiin puede utilizar la palabra clave DBSC (descendente) para aplicar el orden inverso:

```
••lect n....    •ddr•••
f:roa cu•t  r•
order by ∏•-  de•cl
```

Puede ordenar los registros por más de una columna. Además, puede utilizar aliu de columna o incluso sus números de posición (por ejemplo, 3 es la tercera columna de la tabla) en lugar de nombres.

Agrupar y agregar datos

A menudo nos interesa saber cuántas filas contiene un conjunto dado o el valor meélio de una columna (por ejemplo, el promedio en metálico por pedido). MySQL iilc:orpora UD conjunto de funciones de agregación que resultan útiles para iesponder a una consulta. Estas funciones de agregaciónse pueden aplicar a una tabla en su conjunto o a grupos de datos dentro de una tabla. Lo normal es utilizar lu ea.la tabla ID.3

T..... 10.3. Func:ionn de agregacl6n de MySQL.

Nombre	Descripción
AVO(coluimal	Pnlmedlo de loa v81onta de III columnll eapeclftcada.
comrr (eleaentoal	SI espec:iftca una columna, devolvn el nOmero de vmores que no sean NULL de III columrm. SI agrega III palabra DISTIBCT dei.nte del nombre de III columna, obtendré un contador de los valorea diferentes de dicha columna únicemente. SI especifica COUNT (•l, obtendré un recuento de ftlll Independientemente de loe valorea NtJLL.
NIN(columaal	Mlnimo de loa vlllcna de III columna eapec:iftcllda.
MAXC.columna)	Mulmo de loa valorea de lll columna eapec:lftcada.
rm ccoluw)	Denlacl6n eal6ndarde loa valcne de la columna eape,cikNa.
STDDBV(c:oluau)	Iglll que STD(col— _).
SUNCcolumaal	Suma de loa valora de lll columna eapec:lftceda.

Vamos a examinar algunos ejemplos, comenzando por el mencionado anlllrior.,- mente. Podemos calcular la media total de un pedido de la siguiente forma:

```
••lect a,ig(m1111111tl
f:roa orden,
```

El resultado sed el siguiente:

```
| a-.g(-tl    |
|-------------|
|  54.91500:Z |
|-------------|
```

Para obtener información detallada, podemos utilizar la cláusula GROOP BY. Esta cláusula nOS permite ver la media de pedidos por grupos (por ejemplo, por el número de cliente). De esta forma sabremos qué dientes realizan los mayores pedidos.

```
••lect cuet-.rid,  a119(aaount)
erom order•
group by cuet-rid,
```

Al utilizar una cláusula GROUP BY con una función de agregación, se modifica el comportamiento de la función.

En lugar de obtener la media de lu cantidades de pedidos de toda la tabla, esta consulta recupera. Jamedirde las cantidades de pedidos ele cadá cliente (q, pam~ más específico, para cada cuatomerid):

```
| cu•c-.rid | a-.g(aaowitl |
|-----------|--------------|
|     1     |  ..,.,,000:z |
|     :z    |  74.910003   |
|     3     |  47.41500:Z  |
```

Antes de utilizar la agrupación y agregación de funciones, tenga en cuenta que en ANSI SQL, si utiliza una función de agregación o la cláusula GROUP BY, lo único que puede aparecer en la cláusula SBLBCT son lu funciones de agregación y lu columna designadas en la cláusula GROOP BY.

Asf mismo, si quiere utilizar una colwnna en una cláusula GROUP BY, debe listarla en la cláusula SBLBCT.

MySQL brinda una mayor libertad en este sentido ya que admite sintaxis extendida, lo que nos permite dejar elementos fuera de la cláusula SBLBCT si no los deseamos.

Además de agrupar y agregar datos, podemos probar los resultados de una operaci6n de agrepción utilizando la clállUlaBAVIBG&ta •c:oloa. dinctalmnte detrú de la dAmu1a GROUP BY y es como una cláusula llllBRB aplicada dmamente a grupos y agripdc:..

EL_HOLISTICO@HOTMAIL.COM - 2006

Ennuestro ejelllplo anterior. si queremos saber quct clientes tienen pedidos cuyo importe medio supere los SO dólares, podemos utilizar la siguiente consulta:

```
= IÍelKÍ:. ",:u.caÍleild,   a,rgC-C      I
rrom order•
group  i,y  c:uc-rid
baYill'J   a,rg(-CI        > SO¡
```

Fijese en que la cláusula HAVIHGse aplica a los grupos. Esta consulta devolverá el siguiente resulta~o:

```
'-----  -------'--------------'
, 'c:ucii-rill  ¡.:.,rg,-l:l   |
'-------------'-------------+
|          2 |   74.HOOOJ   |
+ '-~··V- ··· ·•1· .-'-'·~--.--•
```

Escé:)gerla;s filas que recuperar

Una cláusula de la instrucción SBLB.CT que puede resultar especialmente útil en aplicaciones Web es la cláusula LIMIT.

Esta cláusula se utiliza para especificar quc! filas deberful devolverse desde el resuitado, Toma dos parámetros: el número de fila desde la que empezar y el número de filas que devolver.

La siguiente consulta ilustra el uso de LIMIT:

```
- ••lec1:-
J~   c;¡ac-r•
umca.  1,   -
```

Es¿ 'c~Uítá se puede l~ como ·s~leécionar nombre de ';lientes y devolver tres mas ell ~l~ul~do, co~do en~ ~a número 2". Fiiese en que los números ~e fila están indexados en cero. es decir. que la primera fila del resultado es la fila númerc>cero.

-isto i:esulta Glllf útil m. aplicaciones Web. Por e¡emplo. cuando el diente está ——e, los productos~eun catilogo y quiere ver 10 por p4P· Sin embargo, la c:1'usula LIMIT no forma parte de ANSI SQL Es una extenaión de MySQL. por lo que al utilizarla el código SQL será incompatible ~ la mayoría de los RDBMS restantes.

Utilizar subconsultas

Una subcomulla es una conaulla anidada dentro de otra consulta: Es una de las novedades de MySQL U. Aunque la mayor parte de su func:iollalldad se puede obtener mediante combinaciones y tablas temporales. lu subconsultas resultan más sencillas de leer y escribir.

EL_HOLISTICO@HOTMAIL.COM - 2006

Subconsultas básicas

La aplic:adón más habitual de una subc:onsulta consiste en utilizar el resultado de ~ consulta para comparar otra. Por ejemplo, si desea buscar el pedido con la cantidad ms alta. podría utilizar la siguiente consulta:

```
..1ecc cu•c011erid,  aeounc
froe order•
wbere aeounc • ‹••l•c:c  max(&II0UDC:  troa order•l;
```

Esta consulta genera el siguiente resultado:

```
'------------'--------+
| cu•coeerid | aeounc |
'------------+--------+
|          4 |  74.98 |
'-------'----'--'------+
```

E~·:ste caso. sólo se devuelve un valor de la subc:onsulta (la cantidad máxima) y se uti11Za para comparar la consulta exterior. Es un buen ejemplo del uso de subco_ns~tas ya que esta consulta en conaeto no se podría reproduc:q:.por medio de combmaaones en SQL ANSI.

Sin embargo. se genera el mismo resultado por medio de esta c:onftlta de combinadón:

```
••lec:l:  cu.coaerid,  amnmc
froe order•
order  by  a.ouac  de•c:
liaic  II
```

En la sentencia de consulta, el número que aparecerá a continuación de que el usuario se salga con de la ala campo de formulario, seguido por una explicativo. Si nuestra si ancia la instrucjcnes ecto, vea que el también de la variable y que el texto que sigue a esta separada por ej cuota | ¡ como ya pueda ver a continuación.

Como depende de LIMIT, esta consulta no es compatible con la mayoría de los RDBMS. aunque se ejecuta con mayor eficacia en MySQL que la subc:onsulta anterior.

Uno de los principales motivos por-los que MySQLno admitía las subc:onsultas es por apenas hay nada que no se pueda hacer sin ellas. Tctcnic:amente se puede crear una sola consulta SQL ANSI correcta con el mismo efecto, pero aplic:ando un truco poco eficaz denominado MAX-CONCAT.

De esta forma se pueden utilizar valores de subconsulta con todos los operadores de comparación convencionales, aunque tambictn existe alguno especial como veremos más adelante.

Subconsultas y operadores

Existen cinco operadores de subc:onsulta especiales. Cuatro se utilizan con subc:onsultas convencionales y el quinto (BXISTS) se suele utilizar con iubconsultas relacionadas, como veremos en un apartado posterior. En la tabla 10.4 se recogen los cuatro operadores de subc:onsulta convencionales.

Tabla 10..4. Operadorw de subconaulta.

[illegible]	[illegible]	[illegible]
ANY	1IHBRB el ,. A11Y (SSLECT el PION t2) ;	eu1 11quieq·cle i. ftlade ˜la subconlllla.
1Il	SBL&c:T el PION t1 1IRDB el 111 (SELEc:T el er- t21;	EquiY8lee•MY.
SOMB	SELEc:T el PROM t1 WIIERB el ,. SOMB ISELECT el ROM t21;!	AllesdeMY:en-ion .. -•mejor.
ALL	SELBc:T el ROM tl WIIERB el ,. ALL (SBL1c:T el tr0III t21;	Devueivetruelilacomp111'8d6nestrueP9'11 toda ... fillsdelasubconsulte.

Todos estos operadoressólo pueden aparecer tras un operador de comparación, con la salvedad de IN que incorpora, por decirlo de algún modo, su operador de comparación(•).

Subconsultas relacionadas

' En una subconsulta relacionada, todo se complica ligeramente. Puede utilizar elementos de la consulta externa dentro de la interna. Por ejemplo,

```
seleec: i8bn, title
   tZ'O!!.——--.
   wbare noc: exists
     . (seleet • tra1l order_items where order_it .... i•tlll•books.isblll;
```

Esta consulta ilustra tanto el uso de subconsultas relacionadas como el uso del úitiino operador de subconsulta especial, EXISTS. Recupera todos los libros que nunca se han pedido (la misma información que devolveríauna combinación por la .izquierda). La consulta interna incluye la tabla order _i tema sólo en la lista FROM perohace referencia a booka. isbn. Es decir, la consulta interna hace referencia a los datos de la consulta externa. &ta es la definición de una subconsulta relacionada. Se buscan filas internas que coincidan (o, en este caso, que no coincidan)con las filas externas.

El operador EXISTS devuelve true si hay filas que coincidan en la subconsulta. Por el contrario, NOT BXISTS devuelve true si no hay filas que coincidan en la subconsulta.

Subconsultas de filas

Huta el momento, todas las subconaultas han devuelto un solo valor, aunque en muchos casos sea true o false (como sucedfa en el ejemplo anterior). Las

subconsultas de filas devuelven una fila completa, que se puede comparar con filas completas de la consulta exterior. Este enfoque se suele utilizar para buscar filas de una tabla que también existan en otra tabla. En nuestra base de datos sobre libros no encontramos un ejemplo pero la sintaxis general podrla ser similar a la siguiente:

```
••leet el, e2, e3
trom tl
wbere (el, e2, e31 ia (seleet el, e2, e3 trom t21;
```

Utilizar una subconsulta como tabla temporal

Podemos utilizar una subconsulta en la cl4usula FROM de una consulta externa. Este enfoque nos permite consultar el resultado de la subconsulta y tratarlo como si fuera una tabla temporal.

La forma más sencilla serfa la siguiente:

```
seleet • trom
(seleet customerid, aaae trom customers where eity•'Box Hill'I
as box_hill_customers;
```

En este caso hemos incluido la subconsultaen la cláusula n0.-. Inmediatamente después de los paréntesis de derre de la subconsulta será necesario asignar un alias a los resultados de la misma. De esta forma podremos utilizarJa como si fuera una tabla cualquiera de la consulta exterior.

Actualizar registros de la base de datos

Además de recuperar datos desde una bue de datos, también resulta normal modificarlos. Por ejemplo, podemos aumentar los precios de los libros incluidos en la base datos. Para ello, podemos utilizar una instrucción Ol>DATE.

La forma habitual de una instrucción Ol>DATBes la siguiente:

```
DPDATB [LOII PR?ORITTI (IGIIORBI nombrec•bJ•
SET eoJumn.ml•expres1dnl,eoJ11111.1•2•e.xpre•i6n2, '..
(IIBBRB concl1c16nl
IORDBR BY er1ter1oe_de_ordenl
[LINIT nO..roJ
```

La idea consiste en actualizar la tabla llamada *nomb~t•bla,* estableciendo cada columna mencionada a la expresión apropiada. Puede limitar la aplicación de la operación Ol>DATB a filas concretas mediante una cláusula WHBRB y restringir el número total de filas con una cláusula LIMIT. ORDER BY sólo se necesita junto a una cláusula LIMIT; si únicamente desea actualizar las 10 primeras filas, puede aplicarles primero algún tipo de orden. Si especific:aLOW_PRIORJ:TYe IGNORE, funcionan de la misma forma que en una instrucdón INSBRT. Vamos a examinar algunos ejemplos. Si desea incrementar los precios de todos los libros un 10 por dento, puede utilizar una instrucción OPDATBcon una cláusula IIHBRE:

EL_HOLISTICO@HOTMAIL.COM - 2006

```
upclace lloek•
..e price•price•l.l;
```

Si quiere cambiar una sola fila (por ejemplo, para actualizar una dirección de cliente), puede hacerlo de la siguiente forma:

```
upclace cu•ca..r•
••e acldr••• ' 'zso '0l••a' Road'
wbere cu•coaerid • 4,
```

Alterar tablas tras su creación

Ademú de actualizar filas, podemos alterar la estructura de las tablas dentro de una base de datos. Para ello, se puede utilizar la instrucción ALTER TABLE. La forma ~sica de esta instrucción es la siguiente:

ALTBR TABLE nombretabla alterac.tdn [, alterac.tdn .•• J

Fijese en que en ANSI SQL sólo puede realizar una alteración por cada instrucción ALTERTABLE, pero MySQL permite realizar todas las que desee. Cada c1'usula de alteración se puede utilizar para cambiar diferentes aspectos de la tabJa.

Si se especifica la d6usula IGNOREe intenta realizar un cambio que genere claves principales duplicadas, la primera se ai\adir6 a la tabla modificada y el resto se eliminarúl. Si no se especifica (el comportamiento predeterminado), la alteración fallar6 y se anular6.

En la tabla 10.5 se recogen los diferentes tipos de alteración que se puede realizar con esta instrucción.

Tabla 10.5. Cambiol posibles con la instniccl6n ALTER TABLE.

ADD (COLONIfl de1cripci6n_colu- (PIRST | APTBR coluDIIIA]: Agrega una nueva columna en la ubicación especificada. Sl no se especifica, la columna inl al final. Fljele an que desc:ripciOn_columna necesita un nombre y un tipo, como en una inslnlCc:lón CUATB.

ADD (C:OLONIIJ (deacripci6n~col11111D&,deacripci6n_columna, •••):Agrega una o varias columnas nuevas at final de la tabla.

ADD IIIIDBX [indice] (colwma, •••): Agrega un Indice a la tabla en la columna o columnas especificadas.

ADD [COIISTRAIIIT [aimbolo) J PRIMARY KBY (columna, •••): Convierte a la columna o columnas especificadas an la clave prlnc:ipal da la tabla. La notación COIISTRAINT u para tablas que ulIIIDn claves HCUndartaa. Lo analizaremo1 en un capltulo posterior.

Sjntaxl: descrjpción

AIID tnlIOUB [COIISTRADT [.S.IlboloJ l [indice] (colwma, •••): All8dl un Indice exc:IUllivo a la tabla en la columna o c:alumn8s especftlc:8du. La notaci6n CONSTJIADITes para tablas InnoOB que utilizan claves secundarias. Lo analw!l'lffl(IS en un capitulo posterior.

ADD (CONSTRAINT [s!mbolo)] l"ORBIGN KBY [indice) (col indice, ••.) [definiciOn referencial: Allade una clave secundarla a una tabla lmoOB. Lo ana~ er1 un capitulo posterior.

ALTBR (C:OLUMN) colwnna (SBT DBFADLT valor | DROP DBPAULTJ: Agrega o elimina un valor predeterminado de una columna dada.

CHANGB [COLOMNI columna descripci6D nueva columna: Cambia la columna llamada columna para que incluya la dIIIC:lipción. Observe que se puede utilizar para cambiar el nombre de la columna porque dHcripci6n_colwma lncluye un nornbnt.

MODIPY [C:OLUMN] dHcripci6n colwma: Similar a CHAJIGB. Se puede utlUzar para cambiar tlpoa de columna, no nombres.

DROP [C:OLUMNJ columna: Elimina columnas c:on nombre.

DROP PRIMARY KBY: Elimina el Indice principal (pero no la columna~

DROP INDBX indice: Elimina el Indice con nombre.

DROP FORBIGIII KBY clave: Elimina la daw secundaria (paro no lw«>lumna).

DISABLB KEYS: Oasactlva la actualización de Indices.

ENABLB KBYS: Activa la actualización de Indices.

RBNAMB [AS) nombre_nuliva_tabla: Cembla el nombre de una tabla.

ORDBR BY nombre col Vuelveacrearlatablaconlasfilasenundeterminadoorden. (Una vez que empiece a cambiar la tabla, las filas dejarén de estar ordenadas.)

CONVBRT TO CHARACTBR SBT ce yCOLLATEc:Convlertetodaslascolumnasbasadas en textoen el conjunto de caractere, -pacificado.

[DEPAULT CIIARACTBR SBT ce y COLLATE e: E1tablece el conjunto de caracteres predeterminado.

DISCARD TABLBSPACIt Elimina el .-chivo de espacio de tabla subyacente para tablas InnoOB (véase un capitulo posterior).

IMPORT TABLBSPACB: Vuelve a crear el archivo de espacio de tabla subyacente para tablas InnoOB (v6aae un capitulo posterior).

table_option1: Le permite restablecerlas opciones da tabla. Utfflza la misma sintaxis que CRBATB TABLB.

Vamos a examinar algunos usos comunes de ALTER TABLE. Algo que suele ocurrir con frecuencia es darse cuenta de que no se ha asignado el ancho suficiente a una columna para albergar los datos. Por ejemplo, en la tabla c:uato-rs, hemos

EL_HOLISTICO@HOTMAIL.COM - 2006

uignado SO caracteres de longitud a los nombres. Al recibir datos, puede que des-_c:µbramosque algunos nombres son demuiadolargos y quedancortados. Podemos solucionar este problema cambiando los tipos de datos de las columnas para aumentar su tama6o a 70 caracteres de largo.

```
altar tabla cuatamera
maclify IIA9e c:har(45) DOC DUll1
```

Algo que tambi6n suele ocurrir de manera habitual es la necesidad de agregar una columna. Imagine que se ha introducido un impuesto local y que Book-0-Rama' necesita agregarlo al total del pedido y realizar su seguimiento por separado. Podemos agrepr una columna para el impuesto(tax) en la tabla ordera de la siguiente forma:

```
alter table ordera
acld tax floatl&,21 after . __t,
```

Otra necesidad que suele surgir de forma habitual es la de eliminar una columna. Podemos suprimir la columna que acabamos de agregar de la siguiente forma:

```
altar table ordera
drop taxl
```

Eliminar registros de la base de datos

La operación de eliminar filas de la base de datos resulta sencilla. Para ello, puede utilizar la instrucción DBLETE, que suele presentar este upecto:

```
DBLrn [LOlf PI.IORIT'l) [QVICIC) [IGIIOU:l PR0II tabla
[IIDU cODdiei6aJ [LINIT mi-ni
[OJUISlt 81' orctea_colu•na•J
(LINIT na.ero)
```

Siesaibe

en solitario, se eliminarán todas las 6las de una tabla, por lo que conviene tener cuidado. Lo normal es eliminar filas concretas, para lo cual se pueden especificar en una cliusula WHBRE. Por ejemplo, podemos eliminar &las si un libro no estuviera disponible o un cliente concreto llevara mucho tiempo si realizar pedidos y quisiera limpiar un poco ~ base de datos.

```
delate fram, cuacl11111ra
llbere C'UatCl111rid-51
```

La cl.iusula LIMIT se puede utilizar para limitar el mlmero m!ximo de &las que se eliminan. ORDBR BY se suele utilizar junto con LIMrr.

LOW_PRIORITY e IGlfORBfuncionan como de costumbre. QUICICpuede resultar mú ripido en tablas MylSAM.

Eliminar tablas

. En ~ones se necesita eliminar una tabla entera. Para ello se puede utilizar la instrucci6n DROP TABLE. Esta operación resulta muy sencilla y pnsenta este upecto.

```
DIIOP TABLS cula1
```

Esta ~trucd6n eliminar, todas las filas de la tab~ y la propia tabla, por lo que es necesano tener cuidado.

Eliminar una base de datos entera

Podemos ir incluso mis al1' y eliminar una base de datos completa con una instrucción DROP DATABASB,que presenta un aspecto parecido al siguiente:

```
DIIOP DAT.IUIIB .INlae de datoa1
```

Esta instrucción elimina todu las filas, todos los índices y la propia base de datos, por lo no es necesario indicar el cuidado que hay que poner al utilizarla.

Lecturas adicionales

En este capítulo, hemos puado revista a los elementos múcómunes-de-SQl.que se utilizan al interactuar con una base de datos MySQL En los siguientes dos capítulos, examinaremos cómo conectar MySQLy PHP para poder acceds a su base de datos desde la Web. También exploraremosalgunas Mc:nicuavanzadas de MySQL Si desea saber mis sobre SQL, puede consultar el esbindar ANSI SQL Lo encontraráen http://www.anai.org.

Si desea obtener mú detalles sobre Ju extensiones de MySQL para ANSI SQL puede consultar el sitio Web de MySQL, en http://www. myaql. com.

A continuación

En el siguiente capítulo veremos cómo permitir el acceso a Book-0-Rama desde la Web.

EL_HOLISTICO@HOTMAIL.COM - 2006

11

Acceder a la base de datos de MySQL desde la Web con PHp

ella, y a filtrar datos de entrada potencialmente peligf090S·

A connnuación, se recogen los aspectos t,ásicos que se abordar6n:

- F~to de las arquitecturas de base de datos Web
- Pasos búiCOS para consultar una base de datos desde la Web
- configurar~ conexión
- Obtener wonnadón sobre bases de datoS cijspc,níble9
- Seleccionaruna baSe de datos que utmzar
- Consultar~ base de datos
- Rec:Uperar los resultados de consulta
- t)esCOt\ectarse de la base de datoS
- Afladir nueva infc,rmaci6na la base de datos

- Utilizar instrucciones predefinidas
- Utilizar otras interfaces de base de datos de PHP
- Utilizar una interfaz genmca de base de datos: PEAK DB

Funcionamiento de las arquitecturas de base de datos Web

En un capitulo anterior se indicaron las lineas búicas del funáonamiento de las arquitecturas de bases de datos Web. A continuación se repiten dichos pasos para ayudarle a recordar:

1. El navegador Web de un usuario envía una petiáón H1TP solicitando una página Web dada. Por ejemplo, el usuario podría soliátar todos los libros escritos por Michael Morgan que tenga Book-0-Rama, utilizando un formulario HI'ML Los resultados de la búsqueda se almacenan en una pAgina denominada reeul ts. php.
2. E~ servidor Web recibe la petición de resulta. php, recupera el archivo *y* lo pasa al motor de PHP para su procesamiento.
3. El motor de PHP comienza a analizar la see12enda dec:emandos.. Dentro di! la secuencia de comandos hay un comando que establece la conexión a la base de datos *y* ejecuta una consulta (realiza la búsqueda de libros). PHP abre una conexión ar servidor MySQL *y* remite la consulta pertinente.....
4. El servidor MySQL recibe la consulta de la base de datos *y* la procesa. A continuación, envía los resultados (una lista de libros) al motor de PHP.
5. El motor de PHP termina de ejecutar la secuencia de comandos, lo que suele implicar la aplicación de formato a los resultados en HTML Seguidamente, devuelve el código HTML resultante al servidor Web.
6. El servidor Web devuelve el código HI'ML al navegador donde el usuario puede ver la lista de los libros solicitados.

Ahora tenemos una base de datos MySQL. por lo que podemos escribir el código PHP para realizar los pasos anteriores. Comenzaremos por el formulario de búsqueda. Se trata de un sencillo formulario escrito en HTML En el listado 11.1 se recap su a\digo:

~ 11.1. M8rd\JIlml. Pllgina de búlqueda de base de datos de Book-0-Rama.

```
clacmb7"•
c.....
    ctitt...-.ill-0•.... C:.talog saarcbc/title>
```



```
c/bead>

cbody>
   cbl>II0ok·O·la.. Catalog Searcbc/bl>

   cfoma action••r••ult•.pbp• ..tbod•"po•t">
      Cbo0•• Searcb Type, cbr ;>
      c•elecc na..•"•••rcbtype•>
         coption valu••"aucbor">Autborc/option>
         coption value•"title">Titlec/option>
         coption valu••"i•bn">ISllllc/option>
      c/•elect>
      cbr />
      lnter S.arcb Teriucbr />
      <input n&me••• ..rcbtera• cype.•text•>
      cbr />
      <input type•"eubait• value.•searcb">
   c/fora>

c/body>
c/btab
```

el !!..J.. de este formulario resulta bastante claro. En la figura ·11.1 se muestra

Figura 11.1. El formulario de búsqueda raulta bastante ... Por lo que se puede bUIC8r un libro a partir de su Utulo, autor o ISBN.

La secuencia de comandos que se inv~rá al pulsar el botón San:h resulta· php. ~ el listado 11.2 se incluye el código completo. En este ca veremos qué realiza esta secuencia de comandos *y* su funci onaauento.

Uamdo 11.2. resultl.php. Ea secuenc:ia de comandos recupe,a loa !'MUitados d IIUSICra base et.datos MySQL Y les aplica lormalo para su visuelzación.

```
cbtal>
cbe&d>
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
ctltle>Book-0- ..... Searcb Re1111t•c/title>
c/baecl>
cb0dY>
cb1>Book-0 · .... searcb Re11111t•</bl >

c?p~~ en• II01lbn• de variabl•• c~rtoe
S•earcbtype•S_IOST['•e&rcbtype,ii
S••rcbtell:99$ IOST('•earcbt-
S•••rcbteza- tr1alt.-.cc:1lt•$81t1

U! ¡¡S•••rc:btype ll 1s-rcbteral
{ ecb0 •Tou ... not entered Marcb detalb. PleaH 'JO bacltand trY ..-111. , ,
ait,

1f (lget_mavic_quote•_gpc<11
{ ecld8lube8($-cbtypel
addml..be•IS•••rcbteral;

• $db • new mymql1('loc:alb0•t' , •boOkora.. '' •bc,okor-123' ' •boOk•' ||
1f (mymqli~cOGDect_errno(11
{ -•d aar. ~t to databAH. P1UH try mgaln later. ';
ecbo ·~-
ait1

•Mlect • fzaml1OOk•wben •.s•earcbtype.• like .,••s~ten.
Sdb ·>C111•rylSqa•ryb

snua_r~••lt•. sr•••lt-~nua_row•1

ecbo •cp>JIUmber of ... foUDd: •.$nua_re•ult8.'</P>'1

for ($1•01 S1csnua_n•ult•, $1++1
{
$row. sre11U1t·>fetcb_a•110C111
ecbo 'cp><•tri 1 • n1•trip81a•bH($row(•dt1•'111
ecbo btal•pK •
ecb0 •c/•troD11>dr />Autbor: '1
•cbo •trip•1..be•ISrow(•autbor')1;
ecb0 •cbr />I , '1
cb0 •triP81•• •ISrow1'i•bD'111
•cbr />Price: ';
ecbo •trip•1•_...•1Srow[•price'II,
ecb0 •c/p>'1
}
sre•alt·>free11,
$db•>C108e0 1

7>
c/b0dY>
c/btab
```


Comprobad que esta secuencia de comandos le permite introducir los caracteres comodín de HTML t y_ (guión bajo), opci6n que puede NSUltarmuy útil para el usuario. Puede escapar estos caracteres si provocan alpn problema en la aplicación. En la figura 11.2 se reproducen los resultados generados al utilizar esta secuencia de comandos para realizar una bdsquedL

Figura 11.2. Los resullados de la bllsqueda de ta bale de datas sobre llllltUs de Java se .,,....., en la pégill8 Web ullllzando la sec:uencie de comandos .....php.

Consultar una base de datos desde la Web

En cualquier sec:uencia de comandos utilizada pára acceder a~ bue d.... desde la Web, se siguen algunos pasos básicos:

1. Comprobar y filtrar los datos procedentes del usuario.
2. Configurar una conexión a la base de datos pertinente.
3. Consultar la base de datos.
4. Recuperar los resultados.
5. Presentar los resultados al usuario.

Éstos 9CIII los pasos seguidos en la secuencia de comandos resulta .php y. vamos a revisar cada uno de ellos por separado.

Comprobar y filtrar datos entrantes

En primer lugar vamos a eliminar todos los esp.cios en blanco que pudiera haber introducido el usuario de manera involuntaria al principio o al final del tmnino de búsqueda.

EL_HOLISTICO@HOTMAIL.COM - 2006

Para ello, aplicamos la fund6n trim (› a S•earchterm.

```
S•earchtera-tri•lS••archterml1
```

El siguiente puo consiste en verificar que el usuario ha introducido un ~ino de blisqueda y un tipo de búsqueda. Tenga en cuenta que eslle puo se realiza tras eliminar los espacios en blanco al final del tmnino de búsqueda. Si hubiáam~ ordenado estas lineas en orden inverso, podrla darse el caso de que el usuano hubiera introducido un tfnnino de blisqueda formado por~~ en ~lanc:o Y.~e esta forma no se generaría un mensaje de error. Por ello, se utiliza la 111Struc:c:lon trimO:

```
if (IS•earchtyp• ll !S••archterml
{ echo 'YOII have not entered ••arch detail•.Plea•• 90 back and try a9ain.'l
  eait1
```

Como observari hemos compropado~a variable Saellrchtype aunque en este caso procede de una instrucción SELECT de HTML. Se estari pr~tando por qué preocuparse de comprobar datos que tienen que introduc:iae. Es importan~ recordar que puede que exista m,s de una interfaz hasta la~~ de datos'. P~r eiemplo, Amazon incluye una gran cantidad de af~iados que utilizan su propaa interfaz de búsqúeda. Ademú, resulta aconsejable filtrar los datos para evatar ~os problemas de seguridad que pueden sur¡ir debido a usuarios procedentes de düerentes puntos de entrada.

Ademú cuando se van a utilizar datos introducidos por un usuano, es lDlportante..filtra;los adecµada~e~te ~ra '!e~l!Cm.!.—lquier carácter de control. En un capitulo anterior, hablamos de las funciones addilla•lieli CT;-strip~lash .. O Y get —gic_quotes_gpc (). Tendr, que escapar datos cuando envae entradas de usuario a una base de datos como MySQL

En este caso comprobamos el valor de la funáón get_-Jic_quotea_~c ‹›' Nos indica si las comillas se aAaden autom'ticamente o no. Sa no se añaden, utilizamos addalash•• () para escapar los datos:

```
if 1!9at_-.gic_quot••-gpc:CII
{
   S.. archtype • add9la•h••IS••archtypel1
   S•earchterm.  add9laah••IS•earchterml1
}
```

También utilizamOII•tripalHhe• O sobre los datos proced':"tes.de la base de datos, Si se ha activado la fundón de comillas m'gicas, los da~ ~~,n barras invertidas al proceder de la base de datos, por lo que sen necesan~ eliminarlas.

Estamoe utilizande>la función ht111111¡1ecidchars (› para codificar caracteres que tengan significados especiales en HTML Nuestros dalm de prueba actuales no incluyen ningún sfmbolo &. menor que(<), mayor que(>) ni c~ dobles(~), pero hay tftulOS de h"bros en los que podrfan aparecer. Esta función nos pemute evitar em>res futurOS-

Configurar una conexión

PHP S incluye una nueva biblioteca para conectarse a MySQL, la biblioteca mysqli. Se puede utilizar con la versión 4 de MySQL y con versiones posteriores. En la versión 4, se ha añadido un nuevo protocolo de conexión mucho m'8 r,pido, que puede utilizar gracias a mysqli. La biblioteca mysqli nos permite utilizar una sintaxis orientada a objetos o basada en procedimientos.

Utilizaremos la siguiente linea en nuestra secuencia de comandos para establecer una conexión al servidor de MySQL:

```
a Sclb • new mysqlil'localhost', 'bookorama', 'bool<or...123', 'boolta'I:
```

Esta función crea una instancia de la clase mysqli y una conexión al host 'localhost', con el nombre de usuario 'bookorama' y la contraseAa 'bookorama12 3'. La conexión se configura para que utilice la base de datos book•.

Al utilizar este enfoque orientado a objetos, podemos invocar métodos en este objeto para ac:c:eder a la base de datos. Si prefiere un enfoque basado en procedimientos, también puede utilizarlo gracias a mysqli. Para ello, utili~ lo siguiente:

```
~ $db • mysqli_connect('localho•t', •booJcorama•, 'bookora..123', •book•'I:
```

Esta función devuelve una puntero en lugar de un objeto, puntero que representa la conexión a la base de datos y, si utiliza el enfoque basado en procedimimtos, tendn que pasarlo en todas las funciones mysqli restantes. Es muy similar al funcionamiento de las funciones de procesamiento de archivos como fopen (›.

La mayoría de las funciones de mysqli cuenta con una interfaz orientada a objetos y una interfaz por procedimientos-. Por lo general, la diferencia es que en la versión por procedimientos los nombres de función comienzan por myaqli_ yes necesario pasar el puntero obtenido por medio de mysqli_ connect (). Las conexiones de base de datos son una excepción a esta regla ya que ae pueden establecer mediante el constructor del objeto mysqli.

Es aconsejable comprobar el intento de conexión ya que si no funciona el resto del código tampoco lo hari. Para ello puede utilizar el siguiente código:

```
if 1,ay•qli_connect_errno(1l
{
   echo 'Error: CO1&ld not connect to databa••' Pleaae try again leter.•;
   exit;
```

(El código es el mismo para la versión orientada a objetos y para la basada en procedimientos.) La función my11qli _connect _errm,f) devuelve un número de error en caso de que se produzca un error o cero en c:ao contrario.

Al conectamos a la base de datos, la línea de código comienza por el operador de supresión de errores, •· De esta forma puede solucionar los enores con elegancia. (También se podría splucionar por medio de excepciones, que no hemos utilizado por tratarse de un ejemplo sencillo.)

EL_HOLISTICO@HOTMAIL.COM - 2006

Hay que tener en cuenta que el número de conexiones simultáneas a MySQL es limitado y vienedetenninado por el parámetromax_connectiona. El objetivo de este parámetro y del parámetro MaxClients de Apache es indicarle al servidor que rechace las solicitudes de nueva conexión en lugar de permitir el uso de todos los recursos del equipo en momentos de mucho tráfico o cuando se haya producido un error en el software.

Los valores predeterminados de estos parámetroS se pueden modificar editando los archivos de configuración. Para establecer el parámetro MaxClient• de Apache, modifique el archivo ht tpd. conf de su sistema. Para establecer el parámetro max_connections de MySQL, edite el archivo my. conf.

Seleccionar una base de datos

Como recordará, al utilizar MySQL desde un interfaz de linea de comandos, necesitamos indicarle qué base de datos tenemos pensado utilizar con la ayuda de un comando como el.siguiente:

```
use book•1
```

Ta,mbién necesitamos realizar esta tarea al establecer la conexión desde la Web. La base de datos que utilizar se especifica como parámetro del constructor mysql i o de la función mysqli_connect (). Si desea cambiar la base de datos predeterminada, puede hacerlo con la función mysqli _select _db (l, a la que puede acceder de dos formas:

o como

```
mysqli_select_db(puacero_bd, aCllllbre_bdl
```

En este caso puede comprobar la similitud entre las funciones que analizamos antes: la versión por procedimientos empieza con myaqli_ y requiere como parámetro ¡adicionalel puntero a la base de datos.

Co.nsultar la base de datos

Para realizar una consulta, podemos utilizar la función mysqli_ query (). Sin embargo, antes de realizar esta tarea, conviene configurar la consulta que se desea ejecutar:

```
$query • •eelect • fi:-c. tiooke vhere •S .. archtype.• lik• •1•.$eearchter111. •1••l
```

En este caso, buscamos el valor introducido por el usuario ($searcbterm) en el campo especificado por el usuario ($searchtype). Como observará, hemos utilizado el operador 1ike en lugar del si al regla general, conviene ser tolerante enS::.~~guuuqsobestablecer la búsqueda (por re bases de datos).

Tenga en cuenta que no es necesario induir un punto Y coma al final de una consulta enviada a MySQL a dififf del monitor de MySQL. ' encia de las consultas realiZadas a través

Ahora podemos ejecutar la consulta:

```
$result.  $db ·>query($query);
```

O, si desea utilizar la interfaz basada en procedl·m·lentos, esta otra v——:

```
$rHult • ,aysqli_query($db, $query|
```

Se le pasa la consulta que se desea ejecutar Y' o pcionalmente, el vínculo a la b.ase de datos (nuevamente, $db).

La versión orientada a objetos devuelve un objeto d lJ da ~ procedimientos un identificador de resultad e ~ta dos ~ enión basa- funaonan las funciones de conexión) En ambos os (similar a 1§orma en que una variable ($result) para utiliza;lo más ad = , el resultado•. almacena en devuelve false. te. En caso de.jllo la fundón

Recuperar resultados de consulta

Existe un conjunto de funciones para divici:. l del objeto de resultados de varias formas El o :' os i:e5ul~dos del identificador o la clave para acceder a las filas devuelt as por a consulta. E lbjeto o identificador de resultados es n nuestro e,emplo, hemos contad 1 d hemos utilizado la función mysqli fe~:h náumero e filas ~evueltas y también tado a objetos, el número de filas d- 1 - seoc (). Al aplicar el enfoque orien- del objeto de resultados , al que puedevue accedto se adlmacenæn el miembro num_rows er e está forma:

```
$num_result• • $result·>num_rovs;
```

Al recurrir a la solución basada en rocedimi .!.. ~ : . : ~ : ~ :: os, ... S. = 70. . . . ., . ~ • 7, :z;t.~7."1

```
$num_reaulte • mysqli_num_rove($resulc|;
```

Resulta útil saber hacerlo. Si tenemos sabremos cuántos hay y podemos proce!::!: ~~~=~°mostrarlos resul~~os,

EL_HOLISTICO@HOTMAIL.COM - 2006

```
for l$i•01  $1 c$aum_re•ult•1  Si++)
{
   11 proce•e  lo•  re•ultado•
```

En cada iteración de este bucle. llamamos a $re•ult->fetch_H•oc () (o a my•qli_fetcb_a•aoc()).

El bucle no se ejecutar6 si no se devuelve ninguna fila. Esta función toma cada fila del conjunto de resultados y devuelve la fila como una matriz, con cada dave en forma de nombre de atributo y cada valor con su valor correspondiente en la matriz:

```
Srov • Sre•ult-»fetch_•••ocll:
```

También puede emplear el enfoque basado en procedimientos:

```
Srov • ayeqli_fetch_•••ocl$reaultl:
```

Dada la matriz asociativa $row, podemos recorrer cada cam~ y mostrarlos adecuadamente,por ejemplo:

```
echo 'cbr />ISU:  •:
echo •trip•l••he•ISrov('i•bn'lll
```

,Como se indicó previamente, hemos llamado a •tripdHbH ‹› para limpiar el valor ante&de IIIOltl'arlo.

Existe una serie de variaciones a la hora de obtener los resultados desde un identificador de resultados. En lugar de utilizar una matriz con claves con nombre, podemos.recuperar..los.resultados.en.una.mab:iz eniunerada.con.my•ql i ... fetcb row ( ) , de la siguiente forma:

```
Srov • Sre•ult-»fetch_rovl$re•ultl1
```

o

```
Srov • 11Y9C1li_fetch_rov(Sre•ultl
```

Los valores de atributo se enumerarán en cada uno de los valores de matriz $row (O], $row (1) etc. (La función myaqli_fetcb_array () nos permite obtener una fila como uno de los dos tipos de matriz o como ambos.)

Podrúimos obtener una fila en un objeto por medio de la funciónmymqli_fetcb _ objecto:

```
SS11W • Sreeult-»fetch_object(l1

~Zt°   ;.
• —~;_, . . u_retch_object(SrHultl,
```

TambWn puede acceder a cada uno de los atributos mediante $row->title, $row--uthor, etc.

Desconectarse de una base de datos

Puede liberar un conjunto de resultados si invoca

```
Sre•ult-»fr . . (l:
```

o

```
-,,.qli_fr.._re•ultl$resultl1
```

Tras ello, podría utilizar

```
Sdb-»clo•e O :
```

o

```
myeqli_clne (Sdbl ,
```

para cerrar una conexión de base de datos. Esta opción no resulta estrictamente necesaria porque se cerrarin de todos modos.cuando una secuencia de comandos finalice su ejecución.

Añadir nueva información a la base de datos

La inserción de nuevos elementos en la base de datos resulta bastante similar a la obtención de elementos de la base de datos. Los pasos son pdcticamente iguales: se establece una conexión,.se env.ía llna ~llllJ.IJ~Y· se comprueban los resultados. En este caso, la consulta enviada sed INSBRT en lugar de SBLBCT. Aunque el proceso es similar, a veces resulta útil examinar un ejemplo. En la figura 11.3, se ilustra un formulario HMTL sencillo para introducir nuevos libros dentro de la base de datos. El código HMTL de esta p6gina se incluye en el listado 11.3.

Listado 11.3. newbook.l'ltml. Código HTML de le pjgin8 de entrad8 de libros.

```
cht•l>
cheacb
   ctitle»aaok-o-a ... - ..., Book lllltryc/title>
c/head>

cbady>
   ch1>8ook-O- .... - ..., Book llltryc/lU>

   cfo:rl1 actiOD•°iaaert  book.php°  ..tbad-•po•t°>
      ctable border•°O°;
         ctr>
            ctd>ISBRc/te.d>
            ctd>ciapu~ type••text• naae••istm• eealength•"ll° •ise•"1:3°~c/
         c/te.r>
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
        ctr>
           ctd>Autborc/td>
           ctd> ciaput type••text• n.me•"•uthor• maxlength•"30"
           ei:r.••"30°:.c/td>
        c/tr>
        ctr>
           ctd>Titlec/td>
           ctd> ciaput type••tezt• n.ae•"title• 11eXleagtla•"I0"
           ei&e•"30°:.c/tcb
        c/tr>
        ctr>
           ctd:.Price $c/td>
           ctd>cinput type-•text• na..••price• mexlengt~•7• ei:•••"7":.c/td>
        c/tr>
        ctr>
           ctd colepen•"2">cinput type•"•ubait• velue•"8-gieter•:.c/td>
        c/tr>
      c/tQle>
   c/fora,-·
c/'oody>
c/btab
```

FIguni 11.3. El pe,sonal de Book-0-Rama puede u11Rzar esta interfaz para la Introducción de nuews Abras en la base de dalos.

Los resultados de este formulario se pasan a insert _book. php, una secuencia de comandos que toma los detalles, realiza algunas operaciones de validación e intenta escribir los datos dentro de la base de datos. El código correspondiente a esta secuencia de comandos se incluye en eJ listado 11.4.

u...do 11.4. Insert_book.php. Esta secuencia de comandos escribe nuevos libros en la base de datos.

r

```
cbtal>
cbeed>
   ctitle>8ook·O-·llaaa aooJc llllltry Reeultec/title,
```



```
c/b..d>
cbody>
cb1>8ook·O·lleae Book Elltry Reeultec/h1>
c?pbp
   // cree DCllbree de verieble cortó•
   $iebn•$ POST['iebn');
   Seutbor;$_JOST[•eutbor')1
   $title•$ IOST['title')1
   $price•$=JOST[price•)1

   if (1$1ebn || !$eutbor || !$title || !$pricel
   (
      echo •You be,,. not ctDtered ;11 tbe required deteile.cbr />'
           ''Pleeee go beck end try ageill.',
      exit;
   } .
   if (19et_11a9ic_quotee....9PC())
   (
      $iebn • edd81..hee($iebnl;
      $eutbor • eddeleebee($eutborl;
      $tiéle • eddel..bee($title);
      $price • doublevel($price);

   • $db • mw myeqli (•locelboet' , 'bookorama', 'bookorama123', 'books');

   if (mysqli_connect_errno()}

      echo •Brror, C-ld not cODDect td'detÜIUle. Ple..• try ~in }!ter.';
      exit;

   Squery • •ia.eert in.to book• veluee
         (•• Sl•bn. ••, '. $autbor. •' , '•. Stitle. ••, '•. Sprice ••J•;
   Srenlt • $db->query($queryl;
   if (Sreeult)
      echo $db-,effected_rowe.• book ia.eerted in.to detebeee:•,

   Sdb·,cloH ();

?>
c/body>
c/btab
```

En la figura 11.4 se muestra el resultado cuando se añade un libro satisfactoriamente. Si examina el código de insert_book .php, ven que en gran parte resulta similar a la secuencia de comandos que escribimos para recuperar datos de la base de datos. Hemos comprobado que todos los campos del formulario se han rellenado y que se les ha aplicado formato correctamente para su inserd6n en la base de: datos con addslashes ():

```
if (!9et_M9ic_quote•_gpc<I I
{
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
$iabn • acld•l..h••ISi•bnl!
$autbor • acldalaabe•(Sauthorl;
$titl• • add•laabea(Stitlel;
$pric• • cloublavall$pric•l1
```

Figura 11.4. L• secuencia de comandos se completa satisfactoriamente e indica que el llbro se ha agregado • la base de datos.

Como el precio se ha almacenado en 1• base de datos como una variable de tipo Rotan te, no queremos colocar barras en su interior. P•r• filtrar todos los caracteres extrai\os que se puedan introducir en este campo numérico, se puede U.mar • la función doubleval (), comentad• en un capítulo anterior. También se encargará de filtrar los simbolos de moneda que pueda haber introducido el usuario en el formulario.

Hemos vuelto a establecer la conexión a la base de datos mediante la creación de una instancia de myaql i y hemos configuradcnma consulta para enviarl• a la bue- de datos. En este caso, la consulta es una instrucción INSBRT de SQL:

```
$query, '•inaert into book• valuea
(' •.Siebn. ••, '• .sauchor.••, '• .$title.•', '•.sprice.••) •;
$reault • $db->query($queryl;
```

&ta se ejecuta sobre la base de datos de la forma habitual llamando a Sdb->query () (o• mysqli_query () si opta por la versión basada en procedimientos). Una diferencia significativa entre utilizar INSERT y SBLBCT está en el uso de mysqli_affected_rows (l. Se trata de una función en la versi~ basa~a en procedimientos y de una variable miembro de una clase en la versión onentada a objetos:

```
echo $db->&ffected_rOW11.' bookl•I inaerted into databaae.•;
```

En la secuencia de comandos anterior, utilizamos mysqli_num_rows () para determinar c:u4ntas Hlas devolvía una instrucción SELBCT. Al escribir consultas que cambien la base de datos como INSBRT, DBLBTE y OPDATB, deberla utilizar en su lugar mysqli _aff ected_rowa () . Con esto, quedan vistos los elementos báicos del uso de bases de datos MySQL desde PHP.

Utilizar .instrucciones predefinidas

La b~blioteca myiqu admi~ el uso de instrucciones predefinidas, que resultan muy ~tiles para ac:eJerar la e,ecución cuando se e;ecutan varias consultas iguales con diferentes da~ T~ mtan los ataques de inyecci6n SQL

El_concepto búico de una mstrucdón predefinida es que se emfa a MySQL una pla~tilla d~ la consulta que queremos ejecutar y, por otra parte, los datos. Puede enviar varios lotes de los mismos datos a la misma instrucci6n predefinida lo q resulta muy útil pua inserciones en bloque. ' ue

En la ——encia decomandos insert_book. php podríamos utilizar instrucciones predefinidas de esta forma:

```
$query • '1Dller1: i111:o boolca value•(?, ?, ?, ?)•;
S•t•t • Sdb->pr•paze(Squ8ry);
$atmt->bind_paraa(•aaad•, $iabn, $author, $title, $price);
S•tiat->execute(I;
echo $atiat->affecc.d_-. ' book iDa•rtÑ i11to databa••'.
$atmt->clo••l1
```

Analicemos el c6dtgo linea a línea.

Al ~ciar ~ co~ta, en lugar de sustituir las variables como hemos hecho antes: m~u~ signos de interrogación para cada &agmento de datos. No debe ai\adir comillas ru otra. delimitadores alrededor de estos signos.

La segunda Una e-la invócaci6n de $db· >prepare e), que se conesparule a mysqli _atmt —repare 1) en la versión basada en procedimientos. Esta lima crea un pun~ro u 0bjeto de instrucci6n que utilizannnos para realizar el procesamiento. ---.E~objeto.de ~~ cuenta COR-un-m.4lodo-llamado bini:t_J1m11 tt. (En la versión ~r procedimientos, se denominaba 11ysqli_etmt_bind_param ().)Este m ét°'.10 indica a PHP qui variables debe sustituir por los signos de interrc>gKióL El pnmer parámetro• una cadena de formato, similar a la utilizada en printf ( ). El valor que pasamos en este caso rsesd•) significa que los cuatro parúnetroe son —·cadena, otra cadena, otra cadena y un doble, respectivamente. También puede u~lizar l~ caracteres~ (entero)o b (para blob). Tras este panmetro debe indicar el nus"?o ?umero de va,:iables que signos de interrogación incluya la instruc:ci611. Se sustttui.rán en el mumo orden. La invocación de $etmt->execute () (mysqli_stmt_exeeate <) en la versión por procedimientos) se encarga de e;e-~tar la consulta. T'8:5 ello.~s acceder a las filas afectadas y cerrar la iJlltruc:-aón. Se preg_untarán esta mstrucaón predefinida es útil o no. Lo importanfees que puede ~~ los valons de las cuatro variables vinculadas y volver a ejec:at.ar la ~~dón sm tena que rehacer el proceso. Resulta muy útil para iterar por mseraones en bloque.

Al igual que los pumaetros, también podemos vincular resultadoa. En~tu d~ tipo SELBCT, puede utilizar use $etmt ->bind_reeul t () (o mysqli at:•t b1nd_reeul t ()) pul ¡m,pon:ionar una lista de variables COR loa que desee com: pletar las columnas de resultados. Cada. vez que se invoque $stat->fetcb () (o

EL_HOLISTICO@HOTMAIL.COM - 2006

ayeqli_etat_fetch (›),los valores de columna de la siguiente fila del conjunto de resultados se c:ompletarút con estas variables Yinc:uJadas. Por ejemplo, en la secuencia de búsqueda de libros que vimos antes, podrfamos utilizar

```
S•t•t•>bind_re8\llt IU911D, Sautllo%, Stitle, Sprice],
```

para vincular estas cuatro variables a las cuatro columnu que se devolverán de la consulta. Tras invocar

```
Setat•>exacutell;
```

puede invocar

```
S•t•t•>fetcbO;
```

en el bucle. Cada vez que se invoque, obtiene la siguiente fila de resultados en las cuatro variables vinculadas. Tambim podemos utilizar myaqli_stmt_bind_ iP•r—l › y,myeqli_a~mt_bind_result (› en la misma secuencia de comandos. -Al cierre de esta edición (PHPSRC2),las instrucciones predefinidas provocaban un fallo de Apache en Windows pero funcionaban correctamente bajo Unix.

Utilizar otras interfaces de base de datos y PHP

PHP admite~ para establecer conexiones a una gran cantidad de bues dedatol, incluidas Oracle,Miamoft SQL Sener y PostgNSQL

Por regla general, los principios para establecer la conexión y consultar cualquiera de las bues de datoe son pdcticamente los mismos. Varían los nombres de Ju funciQnea individualea y las distintas bues de datm tienen funcionalidades j i —t, diferentes, pero si puede establecer una conexi6n a MySQL, no le cos- adaptü esos conocimientos a cualquier situaci6Ji. Si desea utilizar una base de datos que no tenga una biblioteca espedfica disponible en PHP, puede utilizar las : —poBC~ ODBCequivale a Conectividad de bue de datolabierta y es un estúldar para las conexiones a 1,-es de datos. Su funcionalidad es la más restrictiva de cualquier- conjunto, por razones bastante obvias. Si se necesita una com~ completa, nose pueden explotar funciones~

Adidl'as de las bibliotecas incluidas en PHP, tambim estút disponibles clases de "ablltraa::a"'ón de base de datoa como PBAR: eDB, que le permiten utilizar los mismos nmn;hres de función eua cada tipo de base de datos.

Utilizar una interfaz de base de datos genérica: PEAR DB.

Vamos a examinar un breve ejemplo del uso de la capa de abstracd6n PEAR DB. Se trata de uno de los —ponentes fundammtales de PEAR y probablemente el

más utilizado de todos. Si tiene instalado PEAR, ya debería tener el componente DB. De lo contrario, consulte la sección correspondiente del apindice A.

Para permitirle comparar, vamos a ver cómo hubimmos escrito la secuenáa de comandos de resultados de búsqueda utilizando DB.

Uatado 11.5. ruulla_ge,álc.php. Rac:u,,e,.resultadoa de búsquede delde .. bue de datos MySQLy IN aplica fonnato pa,- au vlauallzacl6n.

```
cbtal>
cbeacb
   ctitle>look-O·Rama  Searcb Reeultac/title>
c/bead>
cbody>
cbl>Book·O-llalla Searcb Reeultec/bl>
c?pbp

   // cree nOllbree de variable• corto•
   s..archtype•S_POST('eearcbtype•J;
   Saearcbtera-s_POST['aearcbtera•J;

   Saearcbtera•  tri•($aearcbteral;

   if  l!Saearchtype  11  l$aearcbteral
   {
      ecbo 'You ba•e not entered aearcb detaile. Pleaee 90 bac:Jt &ll!l try again. ':
      exit;


   if  llget_magic_quotea_gpcl11
   {
      S•earcbtype • addalaebea($aearcbtype],
      S•earcbtera  • addalaebeelSeearcbteral;


   // configuraci6n para utilizar  PBAR Da
   require_oncel'DB.pbp':
   s~••r.  ...bookor...':
   Spa••  •  'bookora..123';
   Sbo•ñ  •  'localboet':
   Sclb  ...  ):)ooke  •;

   // configure la cadena de conexi6n universal o DSII
   Sdaa  •  •my•qli://SU•er:Spa•••Sboat/$db_name•;

   // utablesca una coaexi6n a la baae de clatoa
   Sdb • DB::CODDect ($dan,  true],

   // eo11pruebe •i funciona la coaexi6n
   if  IDB::ieBrror]$CUIII
   {
      ecllo  $db->9etNe•ea9e(l1
      exit;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
// realice la conllUlta
tquery• •aelect • fraabaokawhere •.saearcbtype.• lika •1•.SNarcbten.•1••1

SrHult  • Sdb·»qllery(Squeryl ı
// compzuebe que el renltado •• correcto
if CDB::ia&rror(Sreaultl)
(
   echo Sclb·>9etMeaaage e> 1
   uit1


// obtenga el nl'lmero de fila• devuelto
Snua_renlta  • Sreault•>nUIIROwa();

// mueatr• cada fila devuelta
for (Si•0l  Si  csnua_reaultal  Si++I
(
   Srow • Sreault·>fetcllllowCDB_FETCIINDDI_ASSOCI;
   echo 'cp>catrong>'.(Si+ll'.    Title:  ',
   ecbo btalapecialcbare(atripalaabea(Srow(•title'III,
   echo 'c/atrong>cbr•/>Autbor:    '1
   ecbo atripalaabea(Srow(•autbor•J);
   echo •cbr />IS.., ',
   ecbo atripalaabea(Srow(•iabn'II,
   echo •cbr />Price:  '1
   ecbo atripalaataea(Srow(•price•));
   ecbo 'c/p>'1
);

// deacoa6ct••• de la baae de dato•
Sdb·>diaconnect!›,
7>
c;f—ll' •..'''
c/htal>
```

Vamos a examinar los elementos nuevos de esta secuencia de cOIIIUldos.Para ~tableceJ'la conexión utilizamos Ía lfnea

```
$db•  DB::connectl$denl;
```

Esta función acepta una cadena de conexión universal con todos los parúnetros necesarios parar establece!' la conexión a la base de datos. Puede apreciarlo si examina el formato de la cadena de conexión:

```
$dan•  •ayaqli://Suaer:SpaaaeSboat/Sclb_aa1le•1
```

Tras ello, comprobamos si la conexión falló con ayuda del mftodo isBrror (›. y, en caso afirmativo, devolvemos un mensaje de eJ"rOr y salimos:

```
if !DB::ia&rror(Sclbll
(
   echo $clb•>9etMeH&ge ();
   mu.t,
```


Asumiendo que todo haya salido bien, estabJ.ecemosuna comulta y la ejecutamaa de la si~te forma:

```
Srenlt  • Nb·.,...ry($qu9ryl,
```

Podemos mmprobar el nwnero de filas devueltas:

Recaperamo8 cada fila de la siguiente forma:

```
Srow • SnllUlt•>fetcldowCDB_flt▽▽_uaoc:1,
```

Elllitodo genmco fetchRow (› permite recuperar una fila conmuchoeformatos diferentes. El parúnetro DB_FBTCBMODB_ASSOC le indica que queremos recuperar la fila en una matriz asociativa.

Tns mostrar lu filas devueltas, terminamm cerrando la amexión a la bue de datos:

```
$db->disconnect();
```

Como puede apreciar, este ejemplo genérico resulta muy similar a la primera secuencia de comandos.

Las ventajas de usar 08 son que sólo necesitamos recordar un conjunto de funciones de base de datos y que el código requerirá mínimos cambimsi decidimos modificar el software de nuestra base de datos.

Como este libro trata de MySQL, utilizaremos las bibliotecas nativas de MySQL para obteneJ"una mayor velocidad y flexibilidad, pero si lo desea puede utilizar el paquete DBen sus proyectos~~ ~-e~~ma~~~,~!il que~lta~dir úl en detemiinadás 0C.aiiones.

Lecturas adicionales

Si desea obte~er mú información sobre cómo combinar MySQL y PHP, puede leer las secciones pertinentes de manuales sobre PHP y MySQL

Si desea obtener mú información sobre ODBC,visitehttp: 11-. webopedia.com/TERM/0/0DBC.html.

A continuación

En el siguiente capítulo, profundizaremos en el estudio de la administración de MySQL y YeJ"emos formas de optimizar las bases de datos.

EL_HOLISTICO@HOTMAIL.COM - 2006

12

Administración avanzada deMySQL

En este capítulo, vamos a examinar algunos aspectos avanzados de MySQL como los privilegios, la seguridad y la optimización. En concreto abordaremos los siguientes temas:

- Análisis detallado del sistema de privilegios
- Proteger una base de datos MySQL
- Obtener más información sobre la base de datos
- Agilizar los procesos por medio de indices
- Optimizar una base de datos
- Copias de seguridad y recuperación
- Implementar la replicación

Análisis detallado del sistema de privilegios

En un capítulo anterior vimos cómo configurar usuarios y concederles privilegios. Para ello, utilizamos el comando GRANT. Si va a administrar una base de datos MySQL, debe entender la función de este comando y su funcionamiento.

El uso de la instrucción GRANT afecta a las tablas de una base de datos especial llamada mysql. La información sobre privilegios se almacena en cinco tablas de

EL_HOLISTICO@HOTMAIL.COM - 2006

esta bue de datas. Por lo tanto, al conceder privilegi.08sobre bues de datos, debe tener cuidado al otorgar permisos de acceso a la base de datos mysql. El comando GRANT sólo est! disponible a partir de la venión 3.22.11 de MySQL.

Si desea examinar 108 contenidos de la base de datos myaql, regístrese como administrador y escriba

```
U•• ay9ql
```

Tras hacerlo, podrá ver las tablu de la base de datos con la instrucción

```
show tables;
```

como de costumbre.

Los resultados presentarán un aspecto semejante al siguiente:

```
+-----------------+
| Tables_in_mysql |
+-----------------+
| colu—•_prl•     |
| db              |
| func            |
| help_cacegory   |
| help_keyvord    |
| help_relacl-    |
| help_copic      |
| ~o•c            |
| i;Sroc          |
| ~.;JWi•         |
| uaer            |
+-----------------+
```

Cada una de estas tablas almacena información sobre el sistema. Cinco de ellas (uaer, host, clb, tablea_priv, y columna_priv) almacenan información sobre privilegios. Tambiét se conocen como tablu de permisos. Las funciones específicas de estas tablas varían pero todas ellu tiene un mismo objetivo general: determinar lo que los usuarios pueden y no pueden hacer. Estas tablas contienen dos tipos de campos: los campos de ámbito, que identifican al usuario, al host y a la parte de la base de datos a la que hace referencia el privilegio; y los campos de privilegio, que identifican las acciones que puede realizar el usuario en dicho ámbito.

La tablas user y hoat se utilizan para decidir si un usuario puede conectarse al servidor MySQL y si dispone de privilegios administrativos. Las tablas clb y hoat determinan a qué bases de datos puede acceder el usuario. La tabla tablea_pri v determina qué tablas de una base de datos puede utilizar un usuario y la tabla colU111Da_privestablece a qué columnas de la tabla dispone de acceso.

La tabla user

Esta tabla contiene los detalles de los privilegios globales de usuario. Determina si un usuario puede conectane al servidor; MySQL y si dispone de privilegios

globales, es decir, de privilegi08 que se aplican a toda las bases de datos del sis• ma. Para ver la estructura de esta tabla, puede utilizar la instracd6n de•cribe user;. En la tabla 12.1 se recoge el esquema de la tabla uaer.

Tabla 12.1. Esquema de la tat11a user de la bue de C1*S m,aql.

Campo	Tipo
Hoat	varc:har(&Ol
lJaer	varc:har (16)
P•••word	varc:har(41)
Selec:t_priv	enum('11', 'Y' l
Ia.aert_priv	enum('Ir', 'Y' l
lJpdate_priv	enum('N', 'Y')
Delete_priv	enum('N', 'Y')
Create_priv	enum('N', 'Y')
Drop_priv	enum('Ir', 'Y' l
Reload_priv	enuml'N', 'Y' l
Shutdown_priv	enum('N', 'Y')
Proc:eaa_priv	enum('Ir', 'Y' l
Pile_priv	enum('N', 'Y' l
ar-t_priv	enum('N', 'Y')
Referenc:ea_pri v	eDUIDN', 'Y')
Index_priv	enum('N', 'Y')
Alter_priv	enum('N', 'Y')
Show_db_priv	enum('N', 'Y')
Super_priv	enum('Ir', 'Y' l
Create_tmp_table_priv	eaum('11', 'Y')
Loc:k_tablea_priv	enum('11', 'Y')
Exec:ute_priv	enum('11', 'Y')
Repl_alave_priv	enum('11', 'Y')
Repl_c:lieat_pri v	enum('R','Y')
••l_type	enum('11', 'Y')
••l_c:ipher	eDlllll('N', 'Y')
x509_iaauer	enum('ll', 'Y' l

EL_HOLISTICO@HOTMAIL.COM - 2006

x509_subject	enum('11', •y• l
_x_questions	enum('11', 'Y')
max_updates	enum('N', 'y• l
_x_coanections	enu•(•11•, 'y• l

Cada fila de esta tabla se correspondecon un conjunto de privilegios para un usuario procedente de un host y registrado con la contraseñaPassword. Se trata de los campos de ámbito de esta tabla, ya que describen el ámbito del resto de los campos, llamados campos de privilegio.

Los privilegios enumerados en esta tabla (y en las siguientes) se corresponden con los concedidos por medio de la instrucción GRANT en un capitulo anterior. Por ejemplo, Select _pri v se correspondecon el privilegio necesario para ejecutar un comando SELBCT.

Si un usuario tiene un privilegio concreto, el valor de dicha columna será Y. Por el contrario, si a un usuario no se le ha concedido dicho privilegio, el valor será H.

Los privilegios enumerados en la tabla de usuarios son globales, lo que quiere decir que se aplican a todas las bases de datos del sistema (incluida la base de datos mysql). Los administradores tendrán varios valores Y pero la mayoria de los usuarios deberian tener todos H. Los usuarios normales deberian disponer de derechos para acceder a determinadas bases de dates, pero no a todas las tablas.

Las tablas db y host

La mayor parte de los privilegios de los usuarios se almacenan en las tablas db y hoet. La tabla db determina qué usuarios pueden acceder, a qué bases de datos y desde qué host. Los privilegios enumerados en esta tabla se aplican a cualquier base de datos designada en una fila dada. La tabla host complementa a la tabladb. Si un usuario va a conectarse a una base de datos desde varios host, no se enumerará ningún host para dicho usuario en la tabla. En su lugar, se incluirá un conjunto de entradas en la tabla host para especificar los privilegios correspondientes a cada combinación de usuario y host. En las tablas 12.2 y 12.3 se recogen los esquemas de estas dos tablas, respectivamente.

Tabla 12.2. Esquema de la •bla db de la base de datos mysql.

Bost	char(60)
Db	char(64)
IJser	char(16)

Select_priv	enum('lf', 'Y' I
Insert_priv	enum('lf', •y• I
Opdate_priv	enum('lf', 'Y')
Delete_priv	enum('lf', •Y')
Create_priv	enum('lf', 'Y')
Drop_priv	enum('lf', 'Y')
Grant_priv	enua('lf', 'Y'I
References_priv	enum('lf', 'Y' I
Index_priv	enu•('lf', 'Y')
Alter_priv	enum('lf', 'Y' I
Create_tmp_tables_priv	enu•('lf', 'Y' I
Loc::k_tables_priv	enu•('lf', 'Y' I

Tabla 12.3. Esquema de la tabla holt de la bue de datas mysql.

Host.	char(60I
Db	char(UI
Select_priv	enum('lf', •y• I
Insert_priv	enum('lf', •y•)
Opdate_priv	enu•('lf', •y• I
Delete_priv	enum('lf','Y'l
Create_priv	enum('lf', 'Y')
Drop_priv	enum('lf', 'Y')
Grant_priv	enum('lf', 'Y')
References_priv	enum('lf', 'Y'l
Index_priv	enum('lf', 'Y' I
Alter_priv	enum ('lf', 'Y')
Create_tmp_tables_priv	enum('lf', 'Y')
Lock_tables_priv	enum('lf', 'Y'l

EL_HOLISTICO@HOTMAIL.COM - 2006

Las tablas tables_priv y columns_priv

Estas dos tablas se utilizan para almacenar privilegios de nivel de tabla y de nivel de columna, respectivamente. FUlldonan como la tabla db, con la diferencia de que suministran privilegios para las tablas de una base de datos dada y para las columnas de una tabla concreta, respectivamente.

Estas tablas constan de una estructura ligeramente diferente a las tablas user, db y host. En las tablas 12.4 y 12.S se incluyen los esquemas de las tablas tahles_priv y c:olumns_priv, respectivamente.

y.- 12.4. Esquema de .. mlN i.b!NJ)l'iv de la bale d' datos mysql.

H0st	c:har(60)
Db	c:har(64)
	c:har(l6)
Table_na_	c:har(60)
Grantor	c:har(77)
Ti-stamp	ti-stamp(l4l
TÁble_priv	••t (•Selec:t•, •Insert•, •Opdate•, 'Delete•, •c'reate•, •Drop•, 'Grant', •References', •Index•, 'Alter•)
Column_priv	set 1•Selec:t', •In~ert'•- !.tJpclat&.!,--- ---- ----- , Referenc:eii'

Tabla 12.5. Esquema de la tabla columns.JJrivde la base de datos mY9(11.

H08t	c:harl60)
Db	c:har(Ul
0ser	c:har(16l
Table_n•-	c:har(64)
Column_na_	c:har(64)
Ti—•t•!'P	ti—staap(14l
Colu.a_priv	nt (•Selec:t', •Insert•, 'Opdate•, 'Refer-c:es•)

La columna Grantor de la tabla tables _pri v almacena el usuario que concedió el privilegioa este usuario. La columna Timestamp de ambas tablas alm8cena la fecha y la hora en la que se concedió el privilegio.

Control de acceso: cómo utiliza MySQL las tablas de concesión de privilegios

MySQLutiliza las tablas grant para determinarq~ puede hacer un usuario en un proceso de dos fases:

1. Veriflc:ar la c:onexi6n: En esta fase, MySQL comprueba si disponemos de permiso para establecer una conexión en fundón de la informadón de la tabla user, como se indicó anteriormente. Para realizar esta operación, se utiliza el nombre de usuario, el nombre de host y lacontrasei\a. Si se deja en blanco un nombre de usuario, equivaldr6 a todos los usuarios. Los nombres de host se pueden especificar mediante el uso de un cankter comodín (%). Se puede utilizaren forma de campo completo (es dedr, que t equivalga a todos los host) o como parte de un nombre de host, por ejemplo, t. tangledweb. com. au, que equivale a todos los host que terminen en . tang ledweb. com. au. Si se deja en blanco el campo de contraseña, no ser6 necesario utilizar una contrasella. Sin embargo, no conviene utilizar usuarios en blanco, comodines en host, ni usuarios sin contraseñas por cuestiones de seguridad. Si el nombre de hostse deja en blanco, MySQLhace referencia a la tabla host para buscar una entrada user y host que coindda.
2. Veriflc:ar la aolidtad: Cada vez que"ieiñtroifüce úria scilidta4 - -blecer una conexión, MySQL comprueba si disponemos del nivel necesario de privilegios para realizar dicha petición. En primer lugar, el sistema comprueba los privilegios globales (en la tabla user) y si no resultan sufidentes, comprueba las tablas db y host. Si no se dispone de los privilegios necesarios en dicho nivel, MySQL examinan la tabla tahles_priv y, si no son bastan, examinar6la tabla c:olu11s_priv.

Actualizar privilegios: cuándo surten efecto los cambios

El servidor MySQL lee autom6ticamente las tablas de privilegios al iniciarse y al aplicar las instrucciones GRANT y RBVOKE. Sin embargo, ahora que ya sabemos dónde y cómo se almacenai,. dichos privilegios, podemos alterarlos manualmente. Al actualizarlos manualmente,elservidor MySQLno notan que han cambiado.

Es necesario indicarle al servidor que se ha produddo un cambioy existen tres formas de hacerlo. Puede escribir

```
tlu•h privil•9••l
```

EL_HOLISTICO@HOTMAIL.COM - 2006

en la linea de comandos de MySQL (necesitará haber iniciado la sesión como administrador). &ta es la forma mú habitual de actualizar los privilegios.

Tambifn puede ejecutar

```
ay•qladain flu•b ·privileg••
```

o

```
ay•qladain reload
```

desde el sistema operativo.

Tras ello, se comprobar6n los privilegiosglobales cuandD un usuario se conecte, los privilegios de base de datos en la siguiente instrucciónde uso y los privilegios de tabla y columna en la siguiente petición de usuario.

Proteger la base de datos MySQL

La seguridades importante,en especial al establecer la conexión entre la base de datos MySQL y el sitio Web. En esta sección, vamos a repuar las precaudones que se deberíanadoptar para proteger una base de datos.

MySQL desde el punto de vista del sistema operativo

No es aconsejable ejecutar el servidor MySQL (myaqld) como usuario raíz si est6 utilizando un sistema operativo de tipo UNIX ya que permitiría a un usuario de MySQL con un conjunto completo de prjvilegios leer y escribir archivos en cualquier parte del sistema operativo. Es importante tener en cuenta esta dn:unstancia ya que se suele pasar f6dlmente por alto y ya ha sido utilizada para piratear sitios Web basadosen Apache. (AfortuNdamente,se trataba de piratas buenos y lo único que hicieron fue reforzarla seguridad.) Es aconsejable configurar un usuario de MySQL de manera espec:ífic:a para ejecutar mysqld. Ademú puede configurar los directorios (donde se almacenan los datos ffsicamente) para limitar el acceso a dicho usuario. En muchas instalaciones, el servidor se configura para ejecutarse como u•erid my•ql, en el grupo mysql. Tambiin deberla colocar su servidor MySQL tras un cortafuegos. De esta forma podr6 evitar conexiones procedentesde equipos no autorizados. Compruebe si puede conectarse desde el exterior a su servidor a trav& del puerto número 3306; se trata del puerto predeterminadoen el que se ejecuta MySQL y debería estu cerrado en el cortafuegos.

Contraseñas

Asegl1rese de que todos los usuarios tienen contrasei\u (en especial el usuario raíz) y que están bien elegidas y se cambianperiódicamente,como en el caso de las

contraseñas de sistemas operativos. Recuerde que no conviene utilizar contruei\u que contengan palabras o están formadas por palabras de un dicdonario. Lo mejor es utilizar combinacionesde letras y números.

Si va a almacenar contrasei\as en archivos de secuencias de comandos, asegúrese de que sólo el usuario cuya contrasei\a es~ abnac:enada puede verlos.

En las secuencias de comandos de PHP que se utilizan para establecer una conexión a la base de datos, necesitu6 almacenu la- contrasei\a para dicho usuario. Esta operaciónse puede realizar de forma segura colocando eJ nombre de usuario y la contrasei\a en un archivo llamado, por ejemplo, dbconnect. php, que puede situar donde necesite. Esta secuencia de comando&se puede almacenar fuera del 6rbol de documentos Web y configurarla para que sólo resulte accesible al usuario adecuado.

Recuerde que si coloca estos detalles en un archivo . inc o en cualquier otra extensión de archivo en el 6rbol Web, tendr6 que asegúrese de comprobar que el servidor Web sabe que dichos archivos deben interpretarse como PHP, para que los detalles no puedan verse en un navegador Web.

No almacene contrasei\as en forma de texto dentro de la base de datos. Las contrasei\as de MySQL no se almacenan de esa forma, pero resulta habitual en aplicadones Web en las que se desea almacenar los nombres de ufliário y contrue-1\as de los miembros del sitio. Puede ci&ar las c:ontrasei\as utilizando la función SHAl () de MySQL. Recuerde que si ai\ade una contrasei\a coneste formato al ejecutar SELECT (para iniciar sesión), tendr6 que utilizar la misma función de nuevo para COIIIJ*)bar la contrasei\a utilizada por el usuario.

En la parte dedicada a la implementación de proyectos utilizaremos esta funcionalidad.

Privilegios de usuario

Asegúrese de tener claro el sisteina de privilegios de MySQL y las consecuencias de la concesión de los distintos privilegios. No conceda m6s privilegios a un usuario de los necesarios. Para verificarlos, examine las tablas de concesión de privilegios.

En concreto, no conceda los privilegios PROCBSS, PILE, SHUTDOWN y RELOAD a ningún usuario que no sea un administrador a menos que resulte completamente necesario. El privilegio PROCESS se puede utilizar para ver qui estm haciendo o escribiendo otros usuarios, incluidas sus contraseñas. El privilegio FILE se puede utilizar para leer y escribir archivos desde y hasta el sistema operativo (incluyendo, por ejemplo, el archivo /etc/paBSword en un sistema Unix).

El privilegio GRANT debería concederse con p,ecaución ya que permite a los usuarios compartir sus privilegios con otros usuarios.

Asegúrese de que al configurar los usuarios, sólo les concede acceso desde los host que utilizan para conectarse. El usuario j aneelocalhost es correcto pero el usuario jane es bastante común y podría iniciar sesión desde cualquier lugar (y puede que no se trate de la persona que est6 pensando). Evite utilizarcomodinesen los nombres de hosl por razones similares.

EL_HOLISTICO@HOTMAIL.COM - 2006

Puede aumentar mú la seguridad utilizando direcciones IP en lugar de nombres de dominio en la tabla hoat. Esta medida evita errores o intrusos en su DNS. Para ello, inicie el demonio de MySQL con la opción - - skip-name-resol ve, que exige que todos los valores de columna de host sean una dirección IP o un host local.

No debería permitir que los usuarios no administrativos dispongan de acceso al programa mysqladmin en el servidor Web. Este programa se ejecuta desde la linea de comandos por lo que su acceso se convierte en una aspecto relacionado con los privilegios del sistema operativo.

Problemas relaciqnados con la Web

Al conectar la base de datos MySQL a la Web, surgen una serie·de problemas de seguridad especiales.

Conviene, en primer lugar, configurar un usuario especial para las operaciones relacionadas con las conexiones Web. Puede concederle los privilegios mínimos necesario excluyendo DROP,ALTER o CRBATB. Puede concederle el privilegio SBLECT sólo sobre las tablas de catálogos y el privilegio INSERT sobre las tablas de pedidos. Se trata de un nuevo ejemplo de la aplicación del principio de concesión de la me!'or cantidad posible de privilegios.

En el Lltimo capítulo hablamos del uso de las funciones addslashes () y stripslashes () para eliminar caracteres problemáticos en cadenas. Es importante no olvidarse de aplicar estas funciones y de limpiar los datos antas de enviar nada a MySQL. Como recordará, utilizamos la función doubleval (). para corñp-robar que-los dátos riuméricos. erarl. numéricos-. Resulta habitual pasar por alto esta comprobación.

Debería comprobar siempre todos los datos procedentes de un usuario. Aunque el formulario HTML esté formado de cuadros de selección y botones de opción, alguien podría haber modificado el URL para piratear la secuencia de comandos. También conviene comprobar el tamaño de los datos entrantes.

Si los usuarios escriben contraseñas o datos confidenciales para su almacenamiento en la base de datos, recuerde que se transmitirán como texto sin procesar desde el navegador al servidor a menos que utilice SSL (del inglés Secure Sockets Layer, Nivel de sockets seguro). En una sección posterior se ampliará este tema.

Obtener más información sobre bases de datos

Por el momento, hemos utilizado SHOW y DESCRIBE para determinar las tablas que incluye la base de datos y las columnas de las tablas. A continuación, vamos a

analizar otras formas de utilizar estas instrucciones asf como la instrucción BXPLAIN para obtener mú información sobre cómo se realiza una operación de selección-

Obtener información con SHOW

Antes hemos utilizado

```
~bow tablea;
```

para obtener una lista de tablas de la base de datos.

La instrucción

```
•bow databan•;
```

muestra una lista de bases de datos disponibles. A continuación puede utilizar la instrucción SHOW TABLES para ver la lista de tablas de una de esas bases de datos:

```
show tabl•• fr0II booka;
```

Si se aplica la instrucción SHOW TABLES sin especificar una base de datps, se utilizará la base de datos actual de manera predeterminada.

Si sabe cuáles son las tablas, puede obtener una lista de las columnas:

```
show collllllls from order• from booka;
```

Si no activa el parámetro correspondiente a la base de datos, la instrucción SHOW COLUMNS utilizará la base de datos actual. También puede utilizar la notación tabla. columna:

```
show collllll1• from book•.ordera;
```

Existe una variación de la instrucción SHOW para ver los privilegios que tiene un usuario. Por ejemplo, si ejecutamos

```
show graat• for bookora—;
```

se obtiene el siguiente resultado:

```
+-----------------------------------------------------------------------+
| Granu  for bookor•-••                                                 |
+-----------------------------------------------------------------------+
| GRA1ff USAGB Olí •·• to 'bookorama '•'1'                              |
| IDBNTIFIEI> BY PASSWORD '*1BCB&48&41438A2811910DOD7403CS11918BOA85'   |
| GRANT SBLBCT, IIIISBRT, UPDATB, DBLBTB, CRBATB, DROP, I1U>BX, ALTER   |
| OM 'booka'.* to 'bookorama•e'1'                                       |
+-----------------------------------------------------------------------+
```

Las instrucciones GRANT que se muestran no son necesariamente las que se ejecutaron para conceder privilegios a un usuario dado, sino más-bien instrucciones de resumen equivalentes que generan el nivel actual de privilegios del usuario.

EL_HOLISTICO@HOTMAIL.COM - 2006

Existen muchas variantes de la instrucción SHOW. En la tabla 12.6 se recoge un resumen de todas ellas.

Obtener información sobre columnas con DESCRIBE

Como alternativa a la instrucción SHOW COLUMNS, se puede utilizar la instrucción DESCRIBE, que es similar a la instrucción DESCRIBE de Orade (otro RDBMS). Su sintaxis básica es

```
DESCRIBE tabla [columna];
```

Esta instrucción devolver' información sobre todas las columnas de la tabla o sobre una columna dada. Puede utilizar comodines en el nombre de la columna.

Funcionamiento de las consultas con EXPLAIN

La instrucc;ión EXPLAIN se puede utili~r de dos formas. En primer lugar, puede utiliza.-

```
BXPLAINúl•;
```

El resultado de esta instiiic:dóri es tiastiüite similar al devuelto por las instrucciones DESCRIBE tabla o SHOW COLUMNS PROM tabla.

La segunda forma de utilizar BXPLAIN es mucho más interesante ya que permite ver cómo evalúa MySQL una consulta SELECT. Para utilizarla de esta forma, basta con poner la palabra explain delante de una instrucción SELECT.

Puede utilizar la instrucción EXPLAIN cuando esté intentando probar una consulta compleja y el resultado no es el esperado o si tarda más de lo esperado en completarse. Si está escribiendo una consulta compleja, puede probarla antes ejecutando el comando BXPLAIN. Can el resultado obtenido, puede modificar la consulta para optimizarla si resultara necesario. Se trata también de una práctica herramienta de aprendizaje. Por ejemplo, intente ejecutar la siguiente consulta sobre la base de datos Book-0-Rama:

```
explein
••l•ct cuet01Ntre......
f~   c:uat01N1re, ordere, order it.... t,ooke
wbere CU9t0IN1re.cuet0IN1rid • order•.CU8t0IN1rid
aDd ordere.orderid • order ite... orderid
aDd order it.... i•bll • 1,ooke.iebll
aDd book8:title lik• 'tJ•••t;
```


Tabla 12.6. Sintaxis de la instrucción SHOW.

SHOW DATABASES	Enumera las bases de datos disponibles, opcionalmente con nombres como base_de_datos.
SHOW TABLES [FROM base_de_datos]	Enumera las tablas de la base de datos utilizada actualmente o de la base de datos llamada base_de_datos si se especifica. Puede sustituir tabla por el nombre de tablas.
SHOW COLUMNS FROM tabla [FROM base_de_datos]	Enumera todas las columnas de una tabla de la base de datos actualmente en uso o de una base de datos especificada. Puede sustituir columna por nombres de columna. Puede utilizar SHOW FIELDS en lugar de SHOW COLUMNS.
SHOW INDEX FROM tabla [FROM base_de_datos]	Muestra los detalles de todos los índices de una tabla dada de la base de datos actualmente en uso o de otra base de datos llamada base_de_datos si se especifica. También puede utilizar SHOW KEYS en su lugar.
SHOW [GLOBAL|SESSION] STATUS [LIKE estado_elemento]	Devuelve información sobre una serie de elementos del sistema, como el número de subprocesos que se están ejecutando. La cláusula LIKE se utiliza para buscar coincidencias en los nombres de estos elementos; por ejemplo, 'Thread%' recupera los elementos 'Threads_cached', 'Threads_connected' y 'Threads_running'.
SHOW [GLOBAL|SESSION] VARIABLES [LIKE nombre_de_variable]	Muestra los nombres y los valores de las variables del sistema MySQL, como el número de versión. La cláusula LIKE se puede utilizar para buscar correspondencias de forma similar a SHOW STATUS.
SHOW [FULL] PROCESSLIST	Muestra todos los procesos que se están ejecutando en el sistema. La mayor parte de los usuarios verán sus subprocesos pero si disponen del privilegio PROCESS también verán los procesos de todos los demás usuarios, incluidas sus contraseñas si aparecen en las consultas. Las consultas se dividen en secciones de 100 caracteres de manera predeterminada. La palabra clave FULL muestra las consultas completas.
SHOW [GLOBAL|SESSION] STATUS [LIKE estado_elemento]	Muestra información sobre cada una de las tablas de la base de datos actualmente en uso o de la base de datos llamada base_de_datos si se especifica, con la posibilidad de buscar coincidencias con comodines. Entre la información se incluye el tipo de tabla y cuándo se actualizó por última vez.
SHOW [GRANTS FOR usuario]	Muestra la instrucción GRANT requerida para recuperar el nivel actual de privilegio asociado al usuario.
SHOW [GRANTS FOR usuario]	Muestra las instrucciones GRANT necesarias para asignar al usuario especificado en usuario este nivel de privilegios actual.
SHOW [PRIVILEGES]	Muestra los distintos privilegios que admita el servidor.
SHOW CREATE DATABASE base_de_datos	Muestra una instrucción CREATE DATABASE que creará la base de datos especificada.
SHOW CREATE TABLE tabla	Muestra la instrucción CREATE TABLE que creará la tabla especificada.
SHOW [STORAGE] ENGINES	Muestra los motores de almacenamiento disponibles en esta instalación e indica cuál es el predeterminado. (En el capítulo posterior analizaremos los motores de almacenamiento.)
SHOW [STORAGE] ENGINE STATUS	Muestra datos sobre el estado actual del motor de almacenamiento InnoDB.
SHOW [STORAGE] ENGINE LOGS	Muestra información sobre los archivos de registro del motor de almacenamiento BDB.
SHOW WARNINGS	Muestra todos los errores, advertencias o avisos generados por la última instrucción ejecutada.
SHOW ERRORS	Muestra sólo los errores generados por la última instrucción ejecutada.

EL_HOLISTICO@HOTMAIL.COM - 2006

Esta consulta genera el siguiente resultado. (El resultado se muestra verticalmente porque las filas de la tabla superan la anchura del libro. Puede obtener este formato si finaliza la consulta con \Gen lugar de con un punto y coma.)

```
..........................  1. row ..........................
      id, 1
      •elect_type, SIMPLS
      table: .book•
      type, ALL
      po9•i.ble_key81  PRINAaY
      lcey, IIULL
      lcey_leD: IIULL
      ref, IIULL
      rOV•' 4
      Extra: U•ill!I where
..........................  2. rov ..........................
      id, 1
      •elect_type: SIMPLE
      table: order_it...
      type, iDdex
      po98ible_ll:ey8: tRINAaY
      key, tRINARY
      k9Y...len: 11
      ref: IIULL
      rOW•' 4
      Extra: U•ing where; U•ing index
..........................  3. row ..........................
      id, 1
      ••lect_type: SIMPLE
      ta.ble: order•
      type, eq_ref
      pouible_key8, t.RIIIARY..
      key: PRINARY
      key_len, 4
      ref, .book•.order_tt.....orderid
      rOW9: 1
      Extra,
..........................  4. rov ..........................
      id: 1
      •elect_type, SIMPLE
      ta.ble, cu•t — r•
      type, eq_ref
      poHible_key•, PRINAII.Y
      key: PRINARY
  ~ by_len, 4
  ~ ref, .book•.order•. cust-r:Ld
      •• 1
      llxtra,
```

El resultado puede parecer confuso al principio pero resulta muy útil. Vamos a examinar las columnas de esta tabla por separado. La primera columna, id,. indica el número de ID de la instrucción SBLBCT dentro de la consulta a la que hace referencia esta fila. La columna aelect_type explica el tipo de consulta utilizado. El conjunto de valores que puede tener esta columna se recoge en la tabla 12.7.

Tmbl812.7. Posibles tlpoe de HIeccl6n fflOllrados por EXPLAIN.

PRIIIAJIY	(Primera) conaulta externa en la que se ulllizan IUbc:onsultas y uniones.
tnrIOII	Segunda o siguiente consulta de una unión.
DBPBIIDBIIT tJIIIOII	Segunda o siguient9 consulta de una unión, que depende la consulta p,tnc:lpal.
SOBQOIJIY	Subc:onsulta interna.
DBPBIIDBIIT SIJBQODY	Subconsulta Interna, que depende la consulta prlnclpel (H decir, una subconsullli Alfacionacla).
DBRIVBD	Subconsuna utlllzada en la cr•usula PROM.

La columna table enumera simplemente las tablas utilizadas ¡)lira responder a la consulta. Cada fila del resultado ofrece información sobre cómo se ha utilizado dicha tabla en la consulta. En este caso, las tablas utilizadas son ordera, order_items, customers y books.

La columna type explica cómo se est6 utilizando la tabla en las combinaciones de la consulta. En la tabla 12.8 se recogen los distintos valores que puede tener esta columna. Estos valores se enumeran del m6s rápido al mú lento en t4rminos de ~c;:i~ de !a cons~tay propo._rci~ una idea de la cantid8d de ~s_g_ue es necesario leer de cada tabla para ejecutar una consulta.

TNla 12.8. Posibles tipos de comblnaclonH mostradas por EXPLAIN.

	de une sola fila. El tipo •r•te• H utiliza cuando .. trata de una table del sllterna y el tipo con•t en el resto de loa c:asoa.
eq_ref	Para cada conjunm de fllas del NIIO de tables de la combIna-c:ión, leemos una fila de esta lablll. Este tipo H ulllza cuando la combinación usa IDdas las panes del lndice de la tabla y el lndice es exclusivo o n la clave prlnclpal..
ref	Para cada conjunlo de ftlas dll l'HIDde tables de la combinación, leemos un canjunllll de filas de esta table que coinciden en su toálllidad. Esta tipo se utiliza cuando la comblnaclOn no puede 1eleccio,• una 6nica lb en funcl6n. de la condición de combinación, n decir, 11 l6lo H utiliza parte de la clave en la combinación o li no • uclllllva o una clave principal.

EL_HOLISTICO@HOTMAIL.COM - 2006

ref_or_null	Slmhr a una conaulta ref pero MySQL tambl6n busca fllas que sean NtJLL. (Es el tipo m6s ullllzado en subconsult.as.)
inclex_—rge	Se ha utilizado una gpUmización espec:lllca, la combinación de Indicas.
unique_subquery	Esta tipo de combinación se utiliza para sustituir ref en algunas subconsultas IN en las que se devuelve una fila exdusiva.
inclex_subquery	Tipo de comblnaci6nslmNar a unique s\d,query pero qua se utiliza con subc:onsultasindexadas no axc:luslvas.
ranga	Para cada conjunto completo de fll• del resto de las tabl• de la comblnac:i6n. se lee un conjunto de filas de esta tabla induldas en un rango dado.
index	Se examina el Indice completo.
ALL	Se examinan todas las filas de la tabla.

En el ejemplo anterior, puede ver que dos de las tablas se combinas con eq_ref (o~ders y customers), que otra se combina con index (order_items) y que una ter.;era (books) se combina conALL; es decir, examinando todas las filas de la tablas.

La columna rows confirma estas combinaciones: enumera aproximadamente el número de filas de cada tabla que deben examinarse para realizar la combinación. Si multiplica estos valores, obtendr, el número total de 61as que se examinan al llevar a cabo una consulta. Estos números se multiplican porque una combinación es como el producto de las 61as de diferentes tablas (si desea obtener más detalles al respecto, consulte un capítulo anterior). Recuerde que se trata del número de filas ~minado, no el número de filas devuelto y es una estimadón. MySQL no puede saber el número exacto sin llevar a cabo la consulta.

Obviamente, cuanto más pequeAo sea este número mejor. En estos momentos el conjunto de datos de nuestra base de datos resulta insignificante, pero cuando empiece a c:rec:er, el tiempo de ejecución se disparar,. Volveremos sobre este tema en un instante.

La columna possible_keys recoge las claves que MySQL podría utilizar para combinar la tabla. En este caso, se trata de las claves principales.

La columna key es la clave desde la que se utiliza la tabla MySQL o NOLL si no se ha utilizado ninguna clave. Como observar,, aunque existe una posible clave principal para la tabla books, no se utiliza en esta consulta.

La columna key_len indica la longitud de la clave utilizada. Puede utilizarla para determinar si sólo se est4 utilizando parte de una clave. lo cual resulta pertinente cuando se tienen claves compuestas de varias columnas. En este caso, se utilizan las claves completas allí donde se utilizan claves.

La columna ref muestra las colWIUIUutilizadas con la clave al seleccionar filas de la tabla.

Por último, la columna Extra indica cualquier otra información sobre cómo se establece la combinación. En la tabla 12.9 se recogen los valores posibles que pueden aparecer en esta columna.

Tablll 12.9. Valores posiblesde la columna Extra devueltos por la instrucción EXPLAiN.

Valo	Signlficado
Di8tinct	Tras encontrar la primera fila que coincide, MySQL deja de buscar filas.
Not exi•ts	La con.. lta se ha optimizado para utlllzar LBPT JOIN.
Range checked for each record	Intenta buscar el mejor Indice, si existiese, paracadafila del conjuntode filas del resto de las tablas da la comblnacl6n.
tJeing filesort	Selin necesarios doa barrldtia para ordenar los datos. (Por lo tanto, la~ resultar6 el doble de larga.) :f"
tJeing index	Toda la informac:16n de la taia procede del Indice, es decir, no se examlllan las filas.
tJeing temporary	Es necesario crear una talllfi temporal para ejecutar esta consulta. •
tJsing vhere	Se utiliza una dáusula WHBJIB'para selecclonar las filas.

Existen varias formas de solucionar los problemas descubiertos por BXPLAIN. En primer lugar, compruebe los tipos de columnas y asegúrese de que son iguales. En concreto, friese en el ancho de las columnas. Los índices no se pueden utilizar para comparar columnas si tienen diferentes anchos. Para resolver este problema, cambie los tipos de columna de forma que coincidan o intfgrelos en su diseAo desde el principio.

En segundo lugar, puede pedirle al optimizador de combinadones que examine las distribudones de clave y que optimice las combinaciones de forma m,s eficaz por medio de la utilidad myisamchk o la instrucción AIIALYZB TABLE, que son equivalentes. Para invocarla, escriba:

```
myiam.cbk •·all&l.yze ruta_a_la_,,.•e_de_diltoa_iayaql/tabla
```

Puede comprobar varias tablas si las incluye todas en la línea de co~do& o si utiliza:

EL_HOLISTICO@HOTMAIL.COM - 2006

Puede comprobar todas las tablas de todas las bases de datos ejecutando la siguiente linea:

```
myi•a11ebk --uaalyz• r~t•-•l_la_d1rector1o_de_dato•_mysql/•/•.HYZ
```

También puede enumerar las tablas en una instrucdón AHALYZB TABLEdesde el monitor MySQL:

```
analyze table cu•to..:::m: order•, order_it-. boolc•;
```

En tercer lugar, podemos consid.erar la posibilidad de agregar un nuevo Indice a la tabla, si la consulta es a) lenta y b) común. Si se trata de una consulta especial que es probable que no ,-uelva a utilizar, no merecerA la pena el esfuerzo ya que ralentizarA otros procesos. En la siguiente secdón veremos cómo agilizar las consultas.

Agilizar consultas con índices

Si se encuentra en una situadón como la descrita anteriormente, en la que la columnapoHible_keys devuelta por una instrucciónBXPLAINcontiene valo~ NULL, podria mejorar su rendimiento si agrega un índice a la tabla en cuestión. 51 la col~ que está utilizando en la cláusula WHERB resulta idónea para su indexadón, puede-c:reH \UI nuevo-indice paraella con ayuda de la instrucción ALTER TABLE de la siguiente forma:

Optimizar una base de datos

Ademú de los trucos vistos antes para la optimización de consultas, existen otros que permiten inaementar el rendimiento de las bases de datos MySQL

Optimizar el diseño

Es aconsejable que todos los elementos de la base de datos tengan el menor tamaAo posible. Para ello, se debe minimizar la redundancia en la fase de diseño. También se puede logru el mismo objetivo asignando los tipos de datos más pequeAos a las columnas.

También debería reducir al mínimo los valores NOLL y seleccionar una clave principal que sea lo mú corta posible.

Evite el uso de columnas de longitud variable siempre que pueda (como VA..~CHAR, TBXTy BLOB). Si sus tablas tienen campos de longitud fija, resultarúl mú dpidas de utilizar aunque absorban un poco mú de espado.

Permisos

Ademú de aplicar las sugerencias propuestas en la secdón dedicada a BXPLA.IN, puede mejorar la velocidad de las consultas simplificando los permisos. Ya indicamos anteriormente la forma en la que se comprueban las consultas con el sistema de permisos antes de su ejecud6n. Cuanto mú sencillo resulte este proceso, mú rApido se ejecutarAla consulta.

Optimizar tablas

Si lleva utilizando una tabla durante un periodo largo de tiempo, los datos pueden fragmentarse al procesar las actualizaciones y las eliminadones. Como consecuencia, el proceso de búsqueda puede alargarse. Para solucionaz este problema puede utilizar la instrucdón

```
OPTIMIZS TAaLS IICMlbre_de_tabla;
```

o escribir

```
myi —— bk -r table
```

en la línea de comandos.

También puede recurrir a la utilidad myisamchk para ordenar un útdice de tabla y los datos en fundón de dicho índice, de la siguiente forma:

```
myi•amchlc --•ort-index --•ort-record••I
ruca_al_la_d1rector1o_da_dato•_.,,_,i¡•/•.1fYZ
```

Utilizar índices

Utilice índices siempre que lo necesite para aumentar la velocidad de las consultas. Intente que resulten sencillos y no cree índices que sus consultas no vayan a utilizar. Para comprobar los útdices que se estAn utilizando ejecute la instrucción EXPLAINcomo se indicó anteriormente.

Utilizar valores predeterminados

Siempre que le resulte posible, utilice valores predeterminados e ~ dnicamente los datos si difieren de los predeterminados. De esta forma, se redudrA el tiempo que tarda la instruc:ción IRSBRTen ejecutarse.

Otras sugerencias

Existen muchos otros trucos que puede aplicar para mejorar el'rendimiento en situadones concretas y para resolver necesidades particulares. En el sitio Web de MySQLenc:ontrarAuna gran cantidad de ellos. Dirijaseahttp://www.myaq1.com.

EL_HOLISTICO@HOTMAIL.COM - 2006

Realizar una copia de seguridad de la base de datos MySQL

MySQL incorpora dos formas de realizar una copia de seguridad. La primera consiste en bloquear las tablu mientras se copian los archivos físicos, utilizando el comando LOCK TABLES. Su sintaxis es la siguiente.

t.oe1t TABLESsabl• c1po_de_bloqu- (, cabb c1po_de_.bloqueo ••• J

El parámetro *toblo* debería sustituirse por el nombre de una tabla y el *tipo_d,_bloqlllO* deberia ser RBAD o WRITE. Para realizar una copia de seguridad sólosenecesita unbloqueoRBAD.TendráqueejecutaruncomandoPLOSHTABLES; para asegurarse de que los cambios efectuados en los indices se escriben en disco antes de realizar una copia de seguridad.

Los usuarios y las secuencias de comandos pueden ejecutar consultas de sólo lectura durante el volcado. Si tiene un conjunto razonable de consultas q~e alteren la base de datos, como pedidos de cliente, esta solución no resulta práctica.

El segundo método es más avanzado y utiliza el comando myaql_dump. Se suele utilizar de la siguiente forma:

```
ayeqldump --opt ··•ll·databaeee > all.eql
```

Este comando volcará un conjunto de todo el SQL necesario para volver a comtruir la base de datos en el archivo all. aql.

A continuación, deberia detener el proceso mysqld durante un instante y volver a reiniciarlo con la opción --log-update[•archivo_de_registroJ. Las actualizaciones almacenadas en el archivo de registro indicarán los cambios realizados desde la realización del volcado. (Obviamente, debería volcar los archivos de registros en un archivo de copia de seguridad normal.)

Un tercer método consiste en utilizar la secuencia de comandos myaqlhotcopy, que puede invocar de esta forma:

```
ayeqlhotcopy IMlae de datoe /ruta/4-/la/copia de eeguridad
```

Restablecer la base de datos MySQL

Si necesita restablecer una base de datos MySQL, dispondrá nuevamente de dos opciones. Si el problema es una tabla de datos dañada, puede ejecutar myiaamchk con la opción -r {reparar).

Si ha utilizado el primer método de volcado de seguridad, puede copiar los udüvos·dedatos en la misma ubicación dentro de una nueva instalaciónde MySQL

Si ha utilizado el segundo método de volcado, el proceso de restablecimiento comtará de dos fases. En primer lugar, tendrá que ejecutar las consultas en el

a~vo de volcado para reconstruir la base de datos hasta el punto de volcado del archivo. En ——o lugar, d~berá actualizar la base de datos desde el punto almacenado en los archivos de regJStro. Puede hacerlo si ejecuta este comando:

```
11Yaqlbinlog hoatD.11. · bin.(0·tJ• | ayeql
```

Si desea obtener más información sobre el proceso de volcado y recuperación de MySQL, consulte el sitio Web de MySQL en http: //www.myaql. org.

Implementar la replicación

La replicación es una tecnología que nos permite disponer de varios servidores de bases de datos para procesar los mismos datos. De esta forma podemos compartir la carga y mejorar la fiabilidad del sistema; si un servidor deja de funcionar podemos consultar los servidores restantes. Una vez configurada esta posibilidad también se puede utilizar para realizar copias de seguridad.

El ~pto básico consiste en disponer de un servidor principal al que se añad~ d_1ferentessen:icf:oi:e:isecundarios o d!pendientes. Cada uno de éstos duplica al pnnapal Al definir uucialmente los servidores secundarios, copidtos-una instantánea de todos los datos del principal. Tras ello, los secundarios sollitan actualizacioi_- al .l'~cipal, que transmite detalles de las consultas ejeaftiidas desde su registro b~no ~a que los ——darios las vuelvana aplicar a losdatos.

Esta configuración se suele utilizar para aplicar consultas de esctitura al"principal y de lectura a l~ secundarios, operaciones que aplica la lógica de la aplicación. ~~en darse arquitecturas más complejas con varios servidores principales, pero utilizaremos la configuración típica en nuestro ejemplo.

Debe saber que los secundarios no suelen tener datos tan actualizados como los del principal. Esto se da en bases de datos distribuidas.

Para configurar una arquitectura de principal y secundarios, tendrá que asegurarse de que_el registro binario es~ habilitado en el servidor: principal, operación que se descnbe en uno de los apéndices.

Tendrá que editar el archivo my. ini o my. cnf tanto en el servidor principal como en los secundarios. En el principal necesitará estos parámetros:

```
[myaqldl
log·bin
Hrver-id.i
```

El primer parámetro activa el registro binario (que ya deberia haber activado). El segundo asigna al servidor principaJ un ID exclusivo. Los servidores secundarios también necesitan un ID, por lo que añadiremos una línea similar a los archivos my. ini/my. cnf de todos ellos.

——deque se trata de valores exclusivos. Por ejemplo, el ~er servidor secundano podría ~r aerver-id•Z; el siguiente server-id-3; y uf sucesivamente.

EL_HOLISTICO@HOTMAIL.COM - 2006

Configurar el servidor principal

En el servidor principal debe crear un usuario que los secundarios utilizarán pan conectarse. Existe un nivel de privilegios especial para los secundarios denominado secundario de replicación. En función de cómo piense realizar la transferencia de datos inicial, puede que necesite conceder privilegios adicionales de forma temponl. En la mayoría de los casos, se utiliza una instantánea de base de datos para transferir los datos y, en este caso, sólo se necesita el privilegio de replicación especial. Si opta por utilizar el comando LOAD DATA FROM MASTBR para transferir los datos (que analizaremos en el siguiente apartado), este usuario también necesitará los privilegios RBLOAD, SUPER y SELECT,únicamente para la configuración inicial. En cuanto al principio del privilegio menor, descrito en un capitulo anterior, tendrá que revocar estos privilegios una vez configundo el sistema.

Cree un usuario en el servidor principal. Puede asignarle el nombre y la contrasefta que desee, pero no olvide anotar estos datos. En nuestro ejemplo el nombre del usuario es rep_slave:

```
grant replication •lave
OD •.•
to 'r•p-•lave•e•t•  identifi..S by 'contra••da'l
```

&videntemente, puede cambiar la contrasella por otra diferente.

Realizar la transferencia de datos inicial

Puede transferir los datos del servidor principal a los secundarios de v~rias formas. La más sencilla consiste en configurar los secundarios (como veremos en breve) y tras ello ejecutar una instrucción LOAD DATA FROM MASTBR. El problema con este enfoque es que bloquea las tablas del servidor principal mientras se transfieren los datos, lo que puede llevar cierto tiempo, por lo que no es recomendable. (Puede utilizar esta opción si trabaja únicamente con tablas MylSAM.) Por lo general, conviene realizar una instantúlea de la base de datos en el momento actual. Para ello puede emplear los procedimientos que hemos descrito para realizar copias de seguridad. Primero debe vaciar las tablas con la siguiente instrucción:

```
flu•h tal:l1•• with read lock1
```

:: Íj ——n del bloqueo de lectura es que tendrá que registrar el punto en el que se enc:ueiatra el servidor en su registro binario en el momento en que se realiza la instandnea. Puede hacerlo por medio de la instrucción show master stacus;.

Veri un resultado similar al que mostramos a continuación:

Fil•''	Po•ition	aiJalog_Do_Da	ainlog_Ignore_Da
la~ra-ltc-bin.000001	95		

Fijese en File y Pos i ti en, ya que necesitará esta información para configurar los servidores secundarios.

Tras!oo· realice la instantánea y desbloquee las tablas por medio de la siguiente instrucción:

Si utiliza tablas InnoDB, la forma más sencilla consiste en utilizar la herramienta InnoDB Hot Backup, que puede encontrar en el sitio de Jnnobase Oy (ht tp: / /www. innodb. com). No es un programa gratuito por lo que tendrá que adquirir una licencia. También puede realizar el procedimiento que describimos aquf y, antes de desbloquear las tablas, cerrar el servidor MySQL y copiar el directorio completo de la base de datos que desee replicar antes de reiniciar el servidor y desbloquear las tablas.

Configurar el servidor o servidores secundarios

Dispone de dos opciones para configurar el servidor o servidores secundarios. Si ha realizado una instantánea de la base de datos, primero debe .instalarla en el servidor. Tras ello, ejecute las siguientes consultas en el servidor secundario:

```
change ...ter to
ma•t•r·h0tst•'•ervidor',
11&•t•r-u••r••u•uario•,
ma•t•r-pa•8WOrd•'contra•e.la•,
ma•t•r-log-file•'arcJl1vo_regl•tro•,
..•t•r·log·po••po•_regi•tro!
.cart alave;
```

Tendrá que completar los datos mostrados en cursiva. servidor es el nombre del servidor principal. usuario y contraseña provienen de la instrucción GRANT ejecutada en el servidor principal. archivo_registro ypos_regiatro provienen del resultado de SHOW MASTER STATUS tras su ejecución en el serridor principal. Con esto debería funcionar correctamente.

.Si no ha realizado una instandnea, puede cargar los datos desde el serridor principal después de ejecutar la consulta anterior por medio de la siguiente instrucaón:

```
load data froa ma•ter!
```

Lecturas adicionales

Enestos capítulos sobre MySQL, nos hemos centrado en los usos y las partes del sistema más relevantes para el desarrollo Web y en la combinación de MySQL con P~. Si d~ saber más al respecto sobre la administración de MySQL, puede VlSltar el Sitio Web de MySQL en ht tp : / /www.mysql • com.

EL_HOLISTICO@HOTMAIL.COM - 2006

A continuación

En el siguiente capítulo examinaremos algunos de los aspectos avanzados de MySQL que pueden resultar útiles al diseñar aplicaciones Web, como por ejemplo la forma de utilizar los distintos motores de almacenamiento, las transacciones y los procedimientos almacenados.

EL_HOLISTICO@HOTMAIL.COM - 2006

13

Programación avanzada con MySQL

En este capitulo analizaremos determinados aspectos avanzados de MySQL, como por ejemplo los tipos de tabla, las transacciones y los procedimientos almacenados. En concreto, nos detendremos en los siguientes puntos clave:

- La instrucción LOAD DATA INFILE
- Motores de almacenamiento
- Transacciones
- Claves secundarias
- Procedimientos almacenados

La instrucción LOAD DATA INFILE

Una de las características más útiles de MySQLque todavía no hemos analizado es la instrucción LOAD DATA INFILE. Puede utilizarla para cargar datos de tabla desde un archivo. Se ejecuta a gran rapidez. Este flexible comando dispone de numerosas opciones, aunque su aplicación ha~itual suele ser la siguiente:

```
LOAD DATA IIIPILE 'nevbooka.txt'  IIIT0 TABLB bo0ka;
```

Esta línea lee datos de fila del archivo newbooks'txt en la tabla books. De forma predeterminada, los campos de datos del archivo deben separarse mediante

tabuladores y encerrarse entre comillas simples. Cada fila debe separase por una nueva línea (\n). Los caracteres especiales se escapan con una barra invertida (\). Todas estas características se pueden configurar mediante las distintas opciones de la instrucción LOAD, de la que encontrará más detalles en el manual de MySQL.

Para utilizar la instrucción LOAD DATA INFILB, el usuario debe disponer del privilegio FILB, que analizamos en un capítulo anterior.

Motores de almacenamiento

MySQL admite diferentes motores de almacenamiento, en ocasiones denominados tipos de tabla. Esto significa que puede seleccionar la implementadónsubyacente de las tablas. Cada una de las tablas de una base de datos puede utilizar un motor de almacenamiento diferente y resulta muy sencillo convertir de uno a otro.

Puede seleccionar un tipo de tabla al crear una tabla si utiliza

CREATB TABLHAbb TllPB•tipo •.••

Los tipos de tabla posibles son los siguientes

- MylSAM: El tipo predeterminado y el que hemos utilizado hasta el momento en el libro. Se basa en el tipo ISAM tradicional, que equivale a Método de acceso secuencial indexado, un método estándar para almacenar registros y archivos. MyJSAM atlade una serie de ventaju con 1especto aJ tipo ISAM. Si lo comparamos con el resto de motores de almacenamiento, es el que cuenta con mayor mlmero de herramientas para comprobar y .eparar tablas. Las tablas MylSAMse pueden comprimir y admiten búsquedas de texto completas. No son compatibles con transacciones y no admiten claves secundarias.
- ISAM: Se aplica la descripción del tipo anterior. El uso de tablas l5AM ha quedado obsoleto.
- MEMORY (anteriormente conocido como HBAP): Las tablas de este tipo se almacenan en memoria y sus índices se comprimen con hash. Esto las convierte en tablas .ealmente rápidas pero, en caso de que se produzca un fallo, los datos se pierden. Estas características las convierten en la solución ideal para almacenar datos temporales o derivados. Debe espedficar MAX_ROWSen la instrucción CRBATB TABLB o, en caso contrario, estas tablas consumirán toda la memoria existente. No pueden incluir columnas BLOB, TBXT o AUTO IHCRBMBRT.
- MBRGB: Estas tablas le permiten tratar una colección de tablas MylSAM como si se tratara de una sola tabla para poder realizar consultas. De esta fo~ puede solventar limitadones de tamafto múimo en algunos sistemas operativos.
- BDB: Tablas compatibles con transacciones, es decir, cuenbm con opciones COMMITy ROLLBACIC. Son más lentas que las tablas MylSAM pero proporcionan todu las ventajas que supone el uso de transacciones. Se basan en la base de datos Berkeley DB.
- InlloD8: También son compatibles con transacciones y cuentan con las mismas características que las tablas BDB. Admiten claves secundarias. Al ser más rápidas y disponer de más opciones que las tablas BDB, es el motor de almacenamiento compatible con transacciones recomendado.

En la mayoría de aplicaciones Web se suelen utilizar tablas MyJSAMo llUloDB, o una mezcla de ambas.

Conviene utilizar MylSAM cuando en una tabla se .ealizan numerosas operaciones SBLBCT o INSBRT (no una mezcla de las dos) ya que es el tipo más rápido ¡,.ra ello. En muchas aplicaciones Web como catálogos, MyISAM es la mejor opción. Tambim debería utilizarlo si necesita funciones completas de búsqueda de texto. InnoDB se .ecomienda cuando las transacciones son importantes, por ejemplo en tablas en las que se almacenen datos financien» o en casos en los que se entremezclen operaciones INSBRT y SBLBCT, como sucede en foros de mensajes en línea.

Puede utilizar MEMORY con tablas temporales o para implementar vistas, y tabias MERGE si tiene que trabajar con tablas MylSAM de gran ~i\o.

Puede cambiar el tipo de una tabla tras crearla por medio de una instrucción ALTER TABLB, como se indica a continuación:

```
alter table order• cype•ianodb,
alter table order_ite.. cype•1Dnodb!
```

A lo largo del libro hemos utilizado tablas MyJSAM. A continuación nos detend.emos en el uso de transaccionesy en ver cómo se implementanen tablu llUloDB.

Transacciones

Las transaccionesson mecanismos para garantizar la coh~ncia de una base de datos, en especial en caso de que se produzcan enores o un fallo del servidor. En los siguientes apartados analizaremos el concepto de transacción y cómo se implementan con lnnoDB.

Definir transacciones

En primer lugar definiremos el término transacción. Una transacción es una consulta o conjunto de consultu de la que se garantiza su total ejecución en la base de datos o su no ejecución en absoluto. De esta forma, la base de datos conserva un estado coherente independientemente .de la consecución de la transacción.

Para demostrar la importancia de esta posibilidad, imagínese una base de datos de un buco. Imagine que tiene que transferir una cantidad de una cuenta a o~a. Esta operación implica retirar el dinero de una cuenta e ingresarlo en otra, lo que

EL_HOLISTICO@HOTMAIL.COM - 2006

supone al menos dos consultas. Es imprescindible que ambas consultas se ejecuten o que no se ejecute ninguna de ellas. Si retira el dinero de una cuenta y se queda sin luz antes de ingresarlo en la otra, ¿quf sucede? ¿Desaparece el dinero?

Puede que conozca la expresión ACD, un aaónimo inglls que describe los cuatros requisitos que deben cumplir las transacciones:

- Atomicidad: Una transacción debe ser atómica, es dedr, se debe ejecutar por completo o no debe ejecutarse en absoluto.
- Cohenncia: Una transacción debe conservar la base de datos en un estado coherente.
- Aislamiento: Las transacciones sin completar no deben ser visibles para el resto de usuarios de la base de datos, es decir, hasta que se completen deben permanecer aisladas.
- Durabilidad: Una vez escrita en la base de datos, la transacción debe ser permanente.

Una transacción que se haya escrito de forma permanente en una base de datos se denomina confirmada. Una transacción que no se haya escrito en una base de datos, de forma que la base de datos recupere el estado que tenia antes de que se iniciara la transacción, se denomina cancelada.

Utilizar transacciones con lnnoDB

De forma predeterminada, MySQL se ejecuta en modo de confirmación automática, lo que significa que todas las instrucciones que se ejecuten se escriben inmediatamente en la base de datos (se confirman). Si utiliza un tipo de tabla compatible con transacciones, es muy probable que no le interese este comportamiento.

Para desactivarlo en la sesión actual, introduzca lo siguiente:

```
aet autocammit•O,
```

Si el modo de con!irmación automática está activado, tendrá que iniciar una transacción con la instrucción

Si está desactivado, no necesitará este comando ya que se iniciará automáticamente una transac:ck1n al introducir una instrucción SQL Una vez introducidas las iristrucciones que constituyen una transacción. puede confirmarla en la base c;le datos si introduce lo siguiente:

```
— it,
```

Si cambia de opinión, puede recuperar el estado anterior de la base de datos por medio de

```
rollback,
```


Hasta que no confirme la transacción no será visible para el resto de usuarios ni en otras sesiones. Veamos un ejemplo. Ejecute las instrucciones ALTER TABLE del apartado lplterior del capitulo en la base de datos books, como se indica a conti-~uación, en caso de que todavfa no lo haya hecho:

```
alter tabla ordera type•innodb,
altar tabla order_it... type•innodb;
```

Estas instrucciones convierten dos de las tablas en tablas lrínoDB. (Puede volver a convertirlas mú adelante si vuelve a ejecutar la misma instrucción pero con type•MylSAM.) Abra dos conexiones a la base de datos books. En una de ellas, añada un nuevo registro de pedidos a base de datos:

```
injert into ordar• valwt•
(5, 2, ,,.,,. '2004-06-11');
inaert into order ite•• valuea
(5, '0-672-31697-i•. 1);
```

Compruebe si pu~e ver el nuevo pedido:

```
aelect • froa orders wbere orderid•S;
```

Deberla ver el siguiente resultado:

```
+----------+-----------+---------+------------+
| ordarid  | cuatomarid | amount  | date
+----------+-----------+---------+------------+
|        5 |         2 |     91  | 2004·06·11 |
+----------+-----------+---------+------------+
```

Mantenga abierta esta conexión y, en la otra, ejecute la misma consulta select. En ~ta ocasión no verá el pedido:

```
l!llpty aat (0.00 ••el
```

(Si puede verlo, es muy probable que se haya olvidado de desactivar la confirmación automática. Comprufbelo y también si ha convertido la tabla a formato InnoDB.) La razón es que la transacción todavía no se ha confirmado, un buen ejemplo del aislamiento de transacciones en funcionamiento. Vuelva a la primera conexión y confirme la transacción:

```
coaait,
```

Ahora podri recuperar la fila en la otra conexión.

Claves secundarias

InnoDB también admite claves secundarias. Recordar, que analizamos el concepto de clave secundaria en un capitulo anterior. Al utilizar tablu MylSAM, no

EL_HOLISTICO@HOTMAIL.COM - 2006

disponemos de una forma de forzar lu claves secundarias. Imagine, por ejemplo, que aftade una fila a la tabla order_iteme. Tendral que incluir unorderid vallido. Si utiliza MylSAM, tendral que verificar la validez de orderid que dada a la lógica de su aplicación. Al utilizar claves secundarias en InnoDB, puede dejar que la base de datos se encargue de esta comprobación.

Se preguntar6 cómo se configura. Para crear inicialmente la tabla utilizando una clave secundaria, podrfa cambiar la instrucción DDL de la tabla como se indica a continuación:

```
ereate ta!tle order_it..•
(  orderid iat un•I9aed aot aull refereaeee ordere(orderJdl,
   ieba e.bar!lll  not aull,
   quaatity tiayiat uneigaed,
   pri-ry key (orderid, iebn!
   type•IDDOl>B;
```

Hemos aftadido las palabru referencee orders (orderidl tras orderid, lo que significa que esta columna es una clave secundaria que debe incluir un valor de la columna orderid de la tabla ordera.

Por último, hemos ailadido el tipo de tabla type• InnoDB al final de la declaración, necesario para que funcionen lu claves secundariu.

También puede efectuar estos cambios en la tabla existente por medio de instrucciones ALTER TABLE, como se indica a continuación:

```
alter talale ordar_ite•• type•IaaoDB!
alter table order_iteme
add fo~eiga key (orderid)  refereneee ordere(orderidl:
```

Para comprobar si el cambio ha funcionado, intente aftadir una fila con un orderid para el que no exista una fila que coincida en la tabla ordera:

```
ineert into order_it... valuee
!11,   '0-,12-31,11-1',     11,
```

Recibiral un error similar al siguiente:

```
BJ1110R  121,  1230001'  c:annot add or update a ehild  row:
a foreiga key ~oaetra~at fail•
```

Procedimientos almacenados

MySQLS, en su versión alfa al cierre de esta edición. présenta los proc:ectimientos almacenados. Esta versión se convertiral en 2005 en la versión de producción. En los siguientes apartados analizaremos esta c:aracteristica, una de lu principales novedades de la versión S.0.

Un procedimiento almacenado es una función de programación que se crea y almacena en MySQL Puede estar formado por instrucciones SQL y diferentes estructuras de control especiales. Puede resultar muy ótil para realizar la misma

función desde distintas aplicaciones y plataformas, o también como medio para incluir funcionalidad. En una base de datos, los procedimientos almacenados pueden compararse a un enfoque de programación orientado a objetos. Nos permiten controlar la forma de acceder a los datos. Para empezar nos detendremos en un sencillo ejemplo.

Ejemplo básico

En el listado 13.1 se incluye la declaración de un procedimiento almacenado.

Uatado 13.1. baslc_sto'9d_proc:edunt.sql. Declarar un procedimiento almacenado.

```
I Bje11Plo b6eieo de procediaiento al..eenado
deliaiter //

erute procedure total_order• (out total floatl
BIGIII
   eelact •ua(aaount)  into total froa ordare,
!IID
//

deliaiter
```

Analicemos este código linea a lfn~. La primera —6n

```
deliaiter //
```

cambia el valor actual del delimitador final de la instruc:ci6n, que suele ser un punto y coma a menos que lo haya cambiado, por una barra inclinada doble; De esta forma podemos utilizar el delimitador de punto y coma dentro del procedimiento almacenado al alladir código para el mismo sin que MySQL trate de ejecutar el código sobre la marcha.

La siguiente lfnea:

```
eruta procedure total_ord•r•  (out total floatl
```

crea el procedimiento almacenado, que tiene el nombre total_ ordera. Únicamente cuenta con el parámetro total, que es el valor que vallIOIJ a calcular. La palabra OUT indica que este paralmetro se pasa o se devuelve.

Los parámetros también se pueden declarar como IN, lo que significa que el valor se pua al procedimiento, o como IIIOUT, que indica que el valor se pasa pero que el procedimiento lo puede cambiar.

La palabra float indica el tipo del parámetro. En este caso, devolvemos un total de todos los pedidos de la tabla ordera. El tipo de la tabla ordera es float por lo que el tipo devuelto también es float. Los tipos de datos aceptables se asignan a los tipos de columna disponibles.

Si desea mb de ~ paralmetro, puede proporcionar una lista de paralmetios separados por comas como si se tratara de PHP.

EL_HOLISTICO@HOTMAIL.COM - 2006

El cuerpo del procedimiento se enderra entre lu instrucciones BEGIR y BND. Son similares a lu llaves de PHP ({)) ya que delimitan un bloque de instrucciones.

En el cuerpo, simplemente ejecutamos una instrucdón SBLBCT. La única dife.. rencia es que se incluye la cláusula int:o t:ot:al para cargar el resultado de la consulta en el parámetro tot:al. Una vez declarado el procedimiento,se devuelve el delimitador a punto y coma por medio de la línea

```
deliiaiter ;
```

Una vez declarado el procedimiento, se invoca por medio de la palabra clave call:

```
call total_order•(9t);
```

Esta instrucdón invoca los pedidos totales y pasa una variable para almacenar el resultado. Para ver el resultado, tendremos que fijarnos en la variable:

```
Hlect et;
```

El resultado será similar al siguiente:

```
.--'----------------.
| et
.--'----------------.
| i,,.,200111201, |
'-----------------'
```

~ forma similar a la que se crea un procedimiento se puede crear una función. Una función acepta parámetros de entrada (únicamente) y devuelve un solo valor. (Al cierre de esta edición, lu funciones no podían hacer referencia a tablas, pero ~ta limitación cambiará antes de que la versión 5.0 se convierta en la oficial.)

La sintaxis básica de esta tarea es prácticamente la misma. En el listado 13.2 se recoge un ejemplo de función.

U.t8do 13.2. basic_functlon.sql. Dec:l8rar una funci6n ~-

```
| Sintaxi• bA•ica para crear ıı11a f11Dci6n

deli•iter //

creat• fUDCtion add_tax (price floatl return• float
return price•1.11
//

dalia ter |
```

Como puede apreciar, este ejemplo utiliza la palabra clave funct:ion en lugar de procedure. Hay alguna otra diferencia adidonal.

No es necesario especificar los parámetros como IR u OOT, ya que todos son IR, o pañmetros de entrada. Tras la lista de parámetros puede ver la cláusula return• float. Especifica el tipo del valor devuelto. Como antes, este valor puede ser

cualquier tipo válido de MySQL Devolvemos un valor por medio de la instrucción ret:u:rn. cx,mo haríamos en PHP. En este ejemplo no se utilizan instrucdones BEGIR y DI>. Podriamos utilizarlu pero no son necesarias. Al igual que en PHP, si un bloque de instrucciones sólo contiene una instrucción, no es necesario marcar el inicio y el final del mismo. La invocación de una función es un tanto diferente a la de un procedimiento. Puede invocar una función almacenada de la misma forma que se invocaría una función incorporada. Por ejemplo,

```
•alect add_tax(1001;
```

Esta instrucdón deberla devolver el siguiente resultado:

```
.--'-----------.
| add_taxll00I |
'---------------+
            110 |
'--------------'
```

Una vez definidos los procedimientos y las funciones, puede ver el código utilizado para definirlos, si por ejemplo utiliza

```
abow cr•ate proc•dur• total_order•l
```

o

```
•llov cr•at• function addtax;
```

Puede eliminarlos por medio de

```
drop proceclure total_ordar•;
```

o

```
drop function add_tax;
```

Los procedimientos almacenados pueden utilizar estructunis de control, variables, cx,ntroladores DBCLARB (como las excepciones) y un elemento denominado cursor. Lo veremos en los siguientes apartados.

Variables locales

Puede declarar variables globales dentro de un bloque begin .•. end si utiliza una instrucción declare. Por ejemplo, podria modificar la función add_tax para utilizar una variable local para almacenar el tipo impositivo, como se indica en el listado 13.3.

Uatado 13.3. basic_functlons.sql. Dect.,.... une luncl6n 8lmec:en8clll con v•riebles..

```
| Sintaxi• bA•ica para crear ıı11ı fUDC16D

ddi•iter //
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
cr-t.e  f,mcei-   adcl_t.u:  lprice  fl—t.l  nt.urui  float.
begia
   declare  cu: fl-t.  üfault.  0.10,
   ret.una price• ll+t.u:I,
nd
11
delilliclir
```

Como puede comprobar, la variable se declara por medio de declare, seguido por el nombre de la variable y el tipo de la misma. La cl6usula predeterminada es opcional y especifica un valor inicial para la variable. De esta forma puede utilizar la variable como desee.

Cursores y estructuras de control

Veamos un ejemplo mú compte;o. En este caso, definiremos un proc:edimiento almacenado que determine el pedido de mayor importe y devuelva el orderid. (Evidentemente podrfa calcular esta cantidad con una sola consulta, pero este sencillo ejemplo ilustra el uso de cursores y estructura de control.) El código de este procediaúento almacenado se incluye en el listado 13.4.

Ustado 11.4. control_structures_cursors.sqtUUllnr cuno,.. y bucles I)8t8 proceur un conjunto de resulteclos.

```
I Proc:edillient.o para bll•car el pedido ci. ..yor 111PCJrl:e
1 •e podr1a bacer con .ex,  pero  no•  perllit.e ilu•Crar  loe principio•  de  lo•
procediaient.oe~1—cm.sado•
deli•it.er //

creac• procedure. large•t._orde:(oul: largat._id incl
begiD
   declare t.hie id iDt.ı
   declare t.hi•-amount. floac;
   declare 1 a90Ullt. floac ci.faulc 0.0;
   declare 1:id int.;

   declare done int. default. 0;
   declare cont.inue baladler ~or •ql•t.ace •02000• •el: done•  1;
   declare el cureor for •elecc orci.rid, a.ouDl: fro• order•,

   open el,
   repeat.
      fecc:11 el int.o t.bi•_id, t.Jais_a.ouDc;
      if -t. doD• t.beD
         if chi• .aaount. > 1 AIIOIIDI: t..lwll
            .. e i '__—t.•t.hi•__      t
            •el: l_id•t.hi•_id1
         ad  if;
      .. u,
   uncil doa8 eDd repeat.,
   cloee el;
```



```
ud
11

dell.llit.er,
```

Este código utiliza estructuras de control (tanto condicionales como de bucle), cursores y controladores de declaración. Analicmloslo linea a lfnea.

Al principio del procedimiento, declaramos una serie de variables locales para utilizarlas en el mismo. Las variables thia_id y thia_amount almacenan los valores de orderid y ailOunt en la fila actual. Las variables l_amount y l_id almacenan el pedido de mayor importe y su correspondiente ID. Como determinaremos el mayor importe comparando todos los valores con el actualmente mayor, inicializamos esta variable en cero.

Lasiguientevariablededaradaesdone,queinidalizamosencero(falae).Esta variable es el indicador de bucle. Cuando no haya mú filu que analizar, establecemos esta variable en 1 (true).

Lalfnea

```
ci.clare CODl:inue balldler for •ql•t.at.e '02000' •et. done • 1;   -r
```

se denomina controlador de declaración. Es similar a una excepción en procedimientos almacenados. Al cierre de esta edición, se podfan implementar controladores CONTINOB y BXIT. Los primeros, como el que mostramos, recogen la acdón especificada y continúan con la ejecución del procedimiento. Los c:onm>btdcns dv salida salen del bloque begin ... next mú cercano.

La siguiente parte del controlador de declaración especifica cdtllo se invocar6; En este caso, se invocarai al alcanzar aqlatate '02000 •. Se preguntarai qui significa. Significa que se invocm cuando no se encuentien filas. Procesamos el conjunto de resultados fila a fila, y cuando no haya mú filas que procesar, se invoca • este controlador. También podrfamos especificar POR ROT romm. Otras opciones son SQLlfARHIRG y SQLBXCBPTIOR.

El siguiente elemento es un cursor. Es muy similar a una matriz; recupera un conjunto de resultados de una consulta (como la devuelta pormysqli_query (>)y nos permite procesarlo línea a línea (como haríamos, por ejemplo, con myaql i_fetch_row ()). El siguiente cursor:

```
declare el cur•or for •elect. orderid, aeount. fr-  order•;
```

tiene el nombre e 1, una sencilla definición de lo que incluirá. La consulta no se ejecutara itodavía.

La siguiente línea:

```
open c11
```

se encarga de ejecutar la consulta. Para obtener tu filas de datos, debe ejecutar una instrucc:iónfetch,, lo que debe hacer en un bucle repeat. En este caso, el bucle tiene este aspecto:

EL_HOLISTICO@HOTMAIL.COM - 2006

```
repeat

until cloDa ead repear.,
```

F(jese en que la condición no se comprueba hasta el final Los procedimientos almacenados tambim admiten bucles vbile, con la siguiente forma:

```
wllil• coadic:ida do

•ad wbile,
```

Tambim hay bucles loop:

```
loop

ead loop
```

Estos bucles no cuentan con condiciones incorporadas pero se puede salir de los mismos por medio de una instrucción leave;. F(jese en que no hay bucles for.

Siguiendo con el ejemplo, la siguiente línea de código obtiene una fila de datos:

```
rer.c:b el into tbi•_id, tbie_.-ount,
```

Esta linea recupera una fila de la consulta de cursor. Los dos atributos recuperados por la consulta se almacenan en las dos variables locales espedficadas.

,·comprobamos si la fila se ha recuperado y, tras ello, comparamos la cantidad del bucle.actual con la mayor cantidad almacenada, por medio de dos instrucciones IP:

```
1r not cloDa r.ben
   U tbi8_-t     > l -t     tben
      ••t Í — t_-tbi• _____t,
      ••t 1:id•tbi•_id;-
   •ad u,
ead U;
```

Los valores de las variables se establecen por medio de la instrucción set.

Ademú de i f tben, los procedimientos almacenados admiten construcciones i.f .. ·tben else con la siguiente forma:

```
it CODCl1c:1dD tben

    [elNU     c:oadicida tben)

    [elH)

•ad u
```

También existe una instrucción caae, con la siguiente forma:

```
e:...   ftlor
   wll8ia nlor tben inat.ruecida
   [wbeD nl~   tben  Jnatruec:ida••• J
```

```
    [el•• iJlat.ruecidal
ead e•••
```

De nuevo en el ejemplo, una vez terminado el bucle, realizamos las pertinentes laboles de limpieza:

```
el-  el;
••t larv••t_id•l_id;
```

La instrucción e lose derra el cursor.

Por último, establecemos el parámetro OUT en el valor que hemos calculado. No podemos utilizar el par,metro como variable temporal, únicamente para almacenar el valor final. (Es similar a lo que sucede en otros lenguajes de programación como Ada.) Si crea este procedimiento siguiendo esta descripción, puede invocarlo como hicimos con el procedimiento anterior:

```
eall largeet_orderCell;
.Hlect el;
```

Obtendr, un resultado similar al siguiente:

```
.............
| el    |
'-------'
| 3     |
'-------+
```

Puede comprobar personalmente que el c,1cu10es correcto.

Lecturas adicionales

En este capítulo hemos analizado la funcionalidad de los procedimientos alma.,, cenados. En el manual de MySQLencontran m6s infonnación al respecto. Aunque al cierre de esta edición no era muy numerosa, sin duda aumentar, cuando esta versión de MySQL se convierta en la oficial. Si necesita m'5 informadón sobre la. instrucción LOAD DATA INPILE, los distintos motores de almacenamiento-o sobre procedimientos almacenados, consulte el manual de MySQL También puede recurrir a la obra *An lntroduction to* Do,.,,._ Systnu de C.J. O.te, en la que encontrar, má detalles sobre transacciones y cohettflcia de bases de datos relacionales.

A continuación

Hemos desaito los conceptos b'5icos de PHP y MySQL En el sipente capítulo nos detendremos en el comercio electrónico y los aspectos de seguridad relacionados con la configuradón de sitios Web basados en bases de da~.

Parte 111

Comercio electrónico y seguridad

EL_HOLISTICO@HOTMAIL.COM - 2006

14

Crear un sitio Web de comercio electrónico

En este capitulo se presentan algunos aspectos relacionados con la especificación, disei\o, construcción y mantenimiento de un sitio Web de comercio electrónico de manera eficaz. Examinaremos la planificación, los posibles riesgos y varias formas de lograr que un sitio Web resulte autosuficiente.

Examinaremos los siguientes aspectos:

- Decidir q~é se puede conseguir con un sitio de comercio electrónico
- Tipos de sitios Web comerciales
- Riesgos y amenazas
- Seleccionar una estrategia

Decidir un objetivo

Antes analizar los detalles de implementación de su sitio Web, deberla tener claro sus objetivos y disponer de un plan razonablemente pormenorizado para alcanzarlos.

En este libro, partimos del supuesto de que está desarrollando un sitio Web comercial y que, por lo tanto, uno de sus objetivos es ganar dinero.

Existen muchos enfoques comerciales aplicables a Internet. Puede que desee anunciar sus servicios tradicionales o vender en línea un producto del mundo real.

EL_HOLISTICO@HOTMAIL.COM - 2006

Quiz6s tenga un producto que se puede vender y disbibuir por la red. O puede que no tenga intención de utilizar el sitio Web para generar ingresos, sino como medio para fomentar las actividades fuera de linea o como una alternativa menos costosa a las actividades presentes.

Tipos de sitios Web comerciales

Los sitios Web comerciales realizan una o varias de las actividades siguientes:

- Publicar información sobre la compai\fa a travá de folletos en llnea
- Recoger pedidos de artículos o servidos
- Suministrar servicios o artículos digitales
- Ai'ladir valor a los artfculos o servidos
- Reducir costes

Las secciones de muchos sitios Web encajan en varias categorías. A continuación se incluye una descripción de cada una de ellas y la forma habitual en la que generan ingresos u obtienen otros beneficios para una organización. La intención de esta sección del libro es ayudarle a formular sus objetivos. ¿Por qué quiere un sifio-webr¿C&rmva a co11b:ibah cada-fundéR de su sitio Web al negocio?

Medios publicitarios en línea

Pr6cticamente todos los sitios Web comerciales de principios de los ai\os 90 no eran mú que una folleto en l(nea o una herramienta de venta. Este tipo de sitios sigue siendo la forma mú habitual adoptada por los sitios Web comerciales. Ya sea como incursión en la Web o como ejercicio de publicidad de bajo coste, este tipo de sitios sigue teniendo su lógica para muchos negocios.

La presencia publicitaria en Internet puede adoptar distintas formas, desde una tarjeta de visita convertida en p6gina Web a una colección de información publicitaria. En cualquier caso, el objetivo del sitio y la razón financiera de su existencia es atraer a los dientes para que entren en contacto con nuestro negocio.

Este tipo de sitios no genera ningún ingreso directamente, pero puede contribuir a los beneficios obtenidos a través de los medios tradicionales.

El desarrollo de un sitio de este tipo presenta poco& retos téaücos. Los aspectos sobre los que centrarse son similares a los de otros ejercicios de publicidad. Entre los errores m'5 habituales que se suelen cometer al desarrollar este tipo de sitios se incluyen los siguientes:

- No suministrar información importante
- Mala presentación

- No respondera la información generada por el sitio
- No actualizar el sitio
- No rastrear el éxito del sitio

No suministrar información importante

¿Qué es lo que buscan los visitantes que acuden a su sitio? Todo depende de lo que ya aepan. Puede que busqueninfonnación tmlica detallada sobre el produdo o informaci6n muy búica como datos de contacto.

La información que suministran muchos sitios Web no resulta útil o no incluyen los datos mú importantes. Como mínimo, el sitio debe indicar a los visitantes a qué nos dedicamos, en qué 6reas geogrifica del mundo se proporcionan nuestros servicios y cómo pueden ponerse en contacto con nosotros.

Mala presentación

En Internet, una empresa pequetla puede parecer de mayor tamaAo y resultar muy llamativa y, al contrario, una empresa de gran tamaño puede parecer pequeila, poco profesional y poco llamativa si el sitio Web no está bien dilletlado.

Con independencia del tamafto que tenga su empresa, asegúrese de que su sitio Web sea de calidad. El texto debe estar escrito y revisado por alguien con un buen conocimiento del lenguaje correcto. Los gráficos deben ser limpios, deben ser claros y deben tardar poco tiempo en descargarse. En un sitio comercial, debería tener cuidado al elegir los gráficos y los colores de manera que encajen con la imagen qae se desea transmitir. Si utiliza efectos de animación y sorúdo pr~ hacerlo con cuidado.

Aunque no logran que su sitio se genere de la misma forma en todos los equipos, sistemas operativos y navegadores, asegúrese de que utiliza HTML estmdar para que la mayorfa de los usuarios pueda verlo sin errores. Pruébelo con distintas resoluciones de pantalla y con los principales sistemas operativos y navegadores.

No responder a la información generada por el sitio Web

En la Web, un buen servicio de asistencia al cliente resulta tan importante para atraer y mantener clientes como en el mundo exterior. Las compaftfas, ya sean pequei'las o grandes, son responsables de colocar una dirección de correo electr6nico en una página Web y de no examinar o responder a los mensajes de correo r6pidamente.

La gente suele tener diferentes expectativas sobre los tiempos de respuesta a los mensajes de correo electrónico con respedO al correo postal. Si no examina y responde al correo diariamente, los usuarios creer6n que no se concede la importancia necesaria a sus consultas.

Las direcciones que se incluyen en p6ginas Web deberian ser genéricas e ir dirigidas a un puesto o departamento en lugar de a una persona concreta. ¿qué ocurrir6 con el cotreo enviado a fred.smitheejemplo.coatcuando se vaya su

EL_HOLISTICO@HOTMAIL.COM - 2006

titular? Es más probable que el coneo dirigido a ventas•ej emplo. compase a su sucesor. También podría distribuirse a un grupo de personas, lo que contribuiría a garantizar su pronta contestación.

Sin duda recibirá gran cantidad de correo basura en las direcciones que incluya en sus páginas Web. No lo olvide cuando decida cómo reenviar o procesar los correos electrónicos enviados a dichas direcciones. Deberia utilizar información basada en formularios en lugar de proporcionar directamente direcciones de correo electrónico, ya que de esta forma se reduce considerablemente la presencia de correo basura.

No actualizar el sitio

Debe procurar que su sitio Web estt! actualizado. Para ello, es necesario cambiar los contenidos periódicamente. Los cambios en la organización deben reflejarse en el sitio. Un sitio Web con telaraftas desanimará a los visitantes a volver y contribuirá a hacerles creer que gran parte de la información podría ser incorrecta.

Una forma de evitar que un sitio parezca obsoleto consiste en actualizar sus páginas manualmente. Tambit!n puede utilizar un lenguaje de secuencia de comandos como PHP para crear páginas dinámicas. Si su secuencia de comandos dispone de acceso a información actualizada, podrá generar páginas que estén al día de manera constante.

No fealizar el seguimiento del éxito del sitio

Crear un sitio Web es muy buena idea pero, ¿cómo justificar el esfuerzo y el gasto? Uega~á un momento, en especial si se trata del sitio de una gran compañía, en el que se le pedirá que demuestre o cuantifique su valor P.ªra la organización.

En las campañas de marketing tradicionales, las grandes organizaciones invierten mucho dinero en estudios de mercado, antes y desput!s de lanzar la campafta, para comprobar su eficacia. En función de la escala y del presupuesto asignado a la iniciativa Web, estos indicadores podrían resultar igualmente apropiados para el disefto del sitio o para cuantificar su valor.

Entre las opciones disponibles, más caras o más baratas, se incluyen las siguientes:

- Examinar b reptroe del servidor: Los servidores Web almacenan una gran cantidad de datos sobre cada petición realizada al servidor. Gran parte de estos datos resultan de poca utilidad y su formato sin procesar tampoco es de gran ayuda. Para destilar los archivos de registro y obtener un resumen significativo de los datos, es necesario utilizar un analizador de archivos de regiatro. Dos de los más conocidos y gratuitos son Analog. disponible en la dirección http://www. analog. ex, y Webalizer, disponible en este cmqen bttp://www.mrunix.net/webalizer. Tambit!n existen pro- COlllerciales como summary, disponible en ht tp: / / summary. net, qm podrían ";5Ultar más completos. Los analizadores muestran cómo varia el tmlcode 181atio Web con el tiempo y qut! paginas se ven.

- Monitorizar las ventas: El objetivo de los sitios Web con funciones publicitarias es generar ventas. Debería poder estimar el efecto sobre las ventas comparándolo con los niveles de ventas antes del lanzamiento del sitio. Esta operación se complica si se han aplicado otras acciones de marketing en el mismo periodo.
- Solicitar información a loe usuarioe: Si pregunta a sus usuarios, t!stos le dirán lo que piensan del sitio. Puede obtener información útil mediante un formulario o una dirección de correo electrónico con la que recoger los comentarios de los usuarios. Para animar a los usuarios a enviar información puede in~luir un incentivo, como la participación en un sorteo.
- Analizar usuarioe repreaentativoe: La creación de grupos de opinión puede convertirse en una técnica eficaz para evaluar un sitio o incluso en un prototipo del sitio previsto. Para dirigir un grupo de opinión sólo necesitará reunir algunos voluntarios, animarles a valorar el sitio y entrevistarles para obtener y registrar sus opiniones.

  Los grupos de opinión pueden resultar caros si su dirección se delega en moderadores profesionales encargados de valorar y filtrar a los participantes con el objetivo de garantizar una representación demográfica y de personalidad exacta dentro de una comunidad amplia y , a continuación, entrevistar hábilmente a los participantes. Pero el coste de los grupos de opinión tambit!n puede ser nulo, si su dirección se deja en manos de un amateur y si se utiliza un conjunto de personas cuya relevancia con respecto al mercado de destino sea desconocida.

  Una forma de obtener un grupo de opinión bien dirigido mnsiste en recurrir a los servicios de una compañía especializada en investigaciones de mercado, pero no es la única. Si dirige su propio grupo de opinión, procure seleccionar un moderador hábil. Se deben valorar especialmente sus cualidades de comunicación e imparcialidad ante los resultados. Limite el tamaño de los grupos a conjuntos de entre seis y diez personas. El moderador debería estar apoyado por una persona encargada de registrar los datos o por una secretaria que le permita centrarse en la entrevista. La validez de los resultados obtenidos de los grupos se corresponderá con al muestra de gente utilizada. Si utiliza a amigos o a los familiares de sus empleados para valorar el producto, es poco probable que representen a la comunidad general.

Recoger pedidos de artículos y servicios

Si la publicidad del sitio resulta atrayente, el siguiente paso lógico consistirá en permitir que sus clientes realicen el pedido po~Internet. Los pro~i«?nales de las ventas saben que es importante lograr que los clientes tomen una decmón de~ra inmediata. Cuanto más tiempo se conceda al diente para reconsiderar la decisión de compra, mayor será la probabilidad de que mire otras ofertas y cambie de

EL_HOLISTICO@HOTMAIL.COM - 2006

opinión. Si un cliente quiere su producto, Ja mejor estrategia es permitirle realizar la compra de la forma mú fácil y rápida posible. Si obligamos a la gente a desconectar el módem para llamar por teléfono y realizar un pedido o a acudir a una tienda, lo que estaremos haciendo es poner obstáculos a la compra. Si la publicidad en línea ha convencido a un visitante a comprar un producto, permítale que realice la compra de inmediato sin abandonar el sitio Web.

La recogida de pedidos a través de un sitio Web es una buena idea para muchas iniciativas comerciales. Todos los negocios quieren pedidos. La posibilidad de realizar pedidos en linea permite aumentar las ventas o reducir la carga sobre el personal de ventas. Obviamente, tambifn se incurrir6 en deterDJinados costes. La construcción de un sitio Web dinámico, la organización de la infraestructura de pago y el suministro de un servido de atención al cliente suponen gastos. Intente determinar si sus productos resultan adecuados para su comerciali2adón a trav6s de un sitio de comercio electrónico.

Entre los productos que se suelen comprar a trav& de Internet se incluyen libros y revistas, programas y equipos infonmticos, música, ropa, viajes y entradas a eventos de entreteniaúento.

Si su producto no se incluye en ninguna de estas categoriu, no deses—. Estas categorías estúl atestadas de marcas establecidas. Sin embargo, resulta aconsejable considerar alguno de los factores que explican por qué estos productos se venden tan l:!ien por Internet.

De muen ideal, ~ productos de comercio electrónico son imperecederos, sencillos de transportar y lo suficientemente caros como para que los gastos de envios resultan razonables aunque no demasiado caros como para que el comprador sienta la necesidad de examinarlos ffsicamente antes de comprarlos.

Los mejores productos para su comercializacióna trav& de un comercio electrónico son los bienes de consumo. Si quiere -comprar un aguacate, es probable que desee examinarlo visualmente antes e incluso tocarlo. No todos los aguacates son iguales. La copia de un libro, de un CD o de un programa infonmtico suele ser idfntica a otras copias con el mismo tftulo. Los compradores no necesitan ver el artfculo concreto antes de comprarlo.

Ademú, los productos comercializados a trav& de sitios de comercio electrónico deben atraer a la gente que utiliza Internet. En el momento de escribir este libro, este público se compone fundamentalmente de adultos con empleo, jóvenes, con ingresos medios altos y asentados en zonas urbanas. Con el tiempo, sin embargo, los usuarios de Internet abarcarán a toda la población.

Algunos productos nunca aparecerán reflejados en los estudios sobre ventas por Internet. pero son un éxito. Si tiene un producto que atrae únicamente a un nicho del mercado, Internet puede ser un medio fantútico para obtener comparadores.

Algunos productos tienen pocas probabilidades de convertirse en categorías de comercio electrónico. Los artículos baratos y perecederos, como los comestibles, son una mala opción, aunque esto no ha disuadido a las compaMas a intentarlo, aunque sin mucho éxito. Existen categorías que encajan periectamente en sitios publicitarios pero que resultan óptimas para su venta en líneL Por ejemplo, artículos grandes y caros, como vehículos o propiedades inmobiliarias que requieren

mucha investigación antes de su compra, pero que res~tan demasiado caros para servirlos en forma de pedidos y dificiles de transportar.

Existen varios obs~culos para convencer a un compradorpotencial de que reUe,. ne un pedido, a saber:

- Preguntas sin respuesta
- Confianza
- Facilidad de uso
- Compatibilidad

Si un usuario se ve frustrado por cualquiera de estos obs~culos, es probable que abandone el sitio sin comprar.

Preguntas sin respuesta

Si un usuario potencial no encuentra una respuesta inmediata a una pregunta. es probable que abandone el sitio. Este hecho trae consigo una serie de implicaciones. Asegúrese de que el sitio está bien organizado. ¿Resulta sendllo b~r lo que se desea cuando se visita el sitio por primera vez? Asegúrese de que el sitio resulta completo pero sin abrumar a los visitantes. En la Web, los usuarios suelen examinar rápidamente el contenido en lugar de leerlo atentamente, por lo que es aconsejable ser conciso. En la mayor parte de los medios publicitarios, existen limites pr¸ctic;os en cuanto a la cantidad de infonnadml que se puede mdui.r. No oc:une a¡js.(en el caso de un sitio Web. En este caso, los dos lfmites principales son el coste de crear y actualizar la información, y las restricxionesimpuestas por la capacidad para organizar, distribuir y conectar la información sin abrumar a los visitantes.

~ulta tentador imaginar un sitio Web como un comercial que recoge los pedidos autom6ticame"te, que nunca duerme y que no reciben ningún salario. Sin embargo, el servido de asistencia al cliente sigue siendo importante. Anime a los visitantes a formular preguntas. Intente suministrar respuestas inmediatas o pnlcticamente inmediatas por teléfono, por correo electrónico o a trav6s de cualquier otro medio oportuno.

Confianza

Si un visitante no está familiarizado con una marca, ¿por qué debería fiarse de nosotros? Todo el mundo puede aear un sitio Web. La gente no tiene por qué fiarse de lo escrito en el sitio Web y la realización de un pedido requiere una determinada confianza. ¿Cómo distingue el visitante entre una organización con buena reputación o una empresa sin escrúpulos?

A la gente le preocupa una serie de aspectos al comprar en la red:

- ¿Quf se va a hacer con su infarmad6ll penona.17: ¿Se va a vender a otros. se va a utilizar para enviar enormes cantidades de publicidad o se va almact!Tnar en un lugar poco seguro para que otros puedan acceder a ella? Es impor..

EL_HOLISTICO@HOTMAIL.COM - 2006

tante indicar a la gente qué es lo que se va a hacer con sus datos. Es lo que se conoce como política de privacidad y debería resultar f6cil de acceder a ella.

- ¿Su negocio es respetable?: Si su empresa está registrada en una entidad con autoridad relevante, tiene una dirección física y un número de teWfono, y lleva varios aftos funcionando,es menos probable que se trate de un engaño que un negocio compuesto únicamente por un sitio Web y quizás una dirección postal. Asegúrese de mostrar estos detalles.
- ¿Qú ocurre si un comprador no estA satiafecho con su compra?! ¿En qué circunstancias se procederá a realizar un reembolso? ¿Quién se hace cargo de los gastos de envío? Tradicionalmente,las empresas ded1cadas a la venta por correo aplican politicas de reembolso más liberales que los establecimientos tradicionales. Muchos ofrecen una garantfa de satisfacción sin condiciones. Tenga en cuenta los costes de las devoluciones con respecto al aumento de las ventas que generará una politica general de devolución. En cualquier caso, asegúrese de indicar la política aplicada en el sitio.
- ¿Deberfan loe clientes confiarle la lnfonnadón de sus tujetu de Cfflito?: El aspecto más importante relacionado con la confianza de los compradores de Internet es el temor a enviar los datos de su tarjeta de crédito por la red. Por esta razón, debe administrar la información sobre tarjetas de c:r4dito de manera segura y recalcar su preocupación al respecto. Esto significa que, cmno....múlim.Q. debería aplicar el sistema SSL_para transmitir los detalles desde el navegador del usuario al servidor Web y asegurane de que su servidor Web está administrado de manera competente y segura. En una secdón posterior ampliaremos estos temas.

facilidad de uso

Los consumidores varían enormemente en cuanto a su experiencia con los ordenadores, idioma, nivel de alfabetización, memoria y visión. Debe intentar que su sitio resulte lo más sencillo posible de utilizar. Aunque existe una gran cantidad de libros dedicados al diseño de la interfaz de usuario, a continuación se recogen varias diredrices:

- Jntete mantener la sencillez del sitio al múimo: Cuantas más opciones, anuncios o elementos de distracción se incluyan en la pantalla, más probable será confundir al usuario.
- Jntete que el texto resulte claro: Utilice fuentes claras y sencillas. !'''o utilice tamaños demasiado reducidos y tenga en cuenta que en cada equipo se representará con tamai'i.os diferentes.
- ANgdnse de simpllftc:ar al múimo el proceso de ralfzedónde ~dos: La intuición y las pruebas disponiblesavalan la idea de que cuantos mú clics del ratón tenga que aplicar el usuario para realizar un pedido mayor será la posibilidad de que no complete el proceso. Intente reducir al minimo lo

pasos, pero tenga en cuenta que Amazon.com tiene la patente en EEUU sobre un proceso que utiliza un único clic, denominado 1-Click. Esta patente es impugnada enérgicamente por muchos propietarios de sitios Web.

- Procure que los usuuiae no le pierdan: Incluya puntos de referencia y pistas de navegación para indicar a los usuarios dónde se encuentran. Por ejemplo, si un usuario se encuentra dentro de una subsección del sitio, resalte la sei'i.al de navegación correspondiente a dicho sitio.

Si está utilizando la metáfora del carro de la compra en la que se suministra un contenedor en el que los clientes pueden acumular sus compras antes de finalizar la venta, mantenga un vínculo visible al carro en la pantalla constantemente.

Compatibilidad

Asegúrese de probar su sitio en varios navegadores y sistemas operativos. Si el sitio no funciona en un navegador o sistema operativo extendido, el sitio parecerá poco profesional y perderá un porcentaje importante del mercado.

Si su sitio ya es operativo, los registros de su servidor Web pueden indicar qué navegadores utilizan sus visitantes. Como regla general, si prueba su sitio en las dos lltimas versiones de Microsoft Internet Explorer, en una versión reciente de Internet Explorer o Safari en un Apple Mac, en la versión actual de Mozilla para Linux y un navegador ~e sólo text~ ~ ~ynx, su sit!o ~ul~ ará visit,,le para la mayor parte de los usuarios. No olvide probar su sitio con diferentes resoluciones de pantalla. Algunos usuarios utilizan resoluciones muy elevadas pero otros emplean teléfonos o dispositivos PDA. Es complicado que el aspecto de un sitio sea el correcto en una pantalla de 2.048 píxeles de ancho o en otra de tan sólo 240 píxeles.

Intente evitar funciones y mecanismos nuevos, a menos que le apetezca escribir y mantener varias versiones del sitio. El código HTML o XHTML estándar y compatible debería funcionar siempre, aunque funciones más antiguas se admitirán en una mayor variedad de navegadores y dispositivos.

Suministrar servicios y artículos digitales

Muchos productos y servidos se puede vender a través de la Web y enviar a los clientes por correo. Sin embargo,existen servidos que se pueden entregar de manera inmediata a travñ de Internet. Si un servido o artículo se puede trasmitir hasta un módem, se podrá pedir, pagar y entregar de manera instantánea, sin interacción humana. El servido de este tipo más obvio es la información.

En ocasiones la información resulta completamente gratuita o se alimenta de publicidad. En algunos casos la información se suministra a través de una suscripción o se paga por unidades.

Entre los artículos digitales de este tipo se incluyen libros electrónicos y mósica en formato electrónic~ como MP3. Las bibliotecas de imágenes se pueden digitalizar y descargar. El software informático no tiene por qué induine siempre en CD. Se

EL_HOLISTICO@HOTMAIL.COM - 2006

puede comprimir y descargar. Entre los servicios que se pueden vender de esta forma se incluyen el acceso a Internet o servidos de alojamiento Web asf como servicios profesionales que se pueden sustituir por un sistema experto.

El env(o físico de un artículo soliátado desde un sitio Web presenta ventajas e inconvenientes sobre los artículos y servicios digitales. El env(o de un articulo físicamente tiene un coste mientras que las descargas digitales son prkticamente gratuitas. Esto significa que si tiene algo que se pueda duplicar y vender digitalmente, el coste para su sitio será el mismo si vende una unidad que si vende mil. Por supuesto, existen limites: si el nivel de ventas y el tráfico aumenta, se verá obligado a invertir en hardware o ancho de banda.

Los productos o servicios digitales se pueden vender con facilidad como artículos inmediatos. Si una persona realiza un pedido de un articulo ffsico tendrá que esperar uno o varios días para recibirlo. Las descargas se suelen medir en segundos o minutos. La inmediatez puede convertirse en una carga para los comerciantes. Si distribuye un producto digitalmente, tendrá que hacerlo de inmediato. No puede supervisar el proceso manualmente ni distribuir los picos de actividad durante el día. Los sistell\8s de distribución inmediata resultan por tanto más proclives al fraude y suponen una carga sobre los recursos informáticos.

Los artículos y los servicios digitales son ideales para el comercio electrónico pero, como es obvio, el abanico de productos que se puede distribuir de esta forma es limitado.

Añadir valor á los artículos y servicios

Algunas seccionesde sitios Web comerciales no venden ningún articulo ni servicio. Los servidos de seguimientos que incórpoün las compaMas de merisajeria (UPS en vww. upe.como Fedex en www. fedex. com) no están diseñados para generar ingresos directamente. Aftaden valor a los servidos existentes ofrecidos por la organización. Las compaftlas obtienen una ventaja competitiva al permitir que los clientes puedan saber dónde se encuentra su paquete o acceder a sus cuentas bancarias. Dentro de esta categoría se incluyen los foros de ayuda. Existen fundadas razones comerdales para ofrecer a los clientes un área de debate en la que compartir sugerencias para resolver problemas sobre los productos de su compañía. Los clientes pueden resolver sus problemas consultandolas soluciones dadas a otros, los clientes internacionales puede obtener ayuda sin tener que realizar llamadas internacionales y pueden responder a preguntas de otros clientes fuera de los horarios de oficina. Este tipo de servicios puede contribuir a aumentar la satisfacción de los clientes con un coste muy bajo.

Recortar costes

Internet se suele utilizar para recortar costes. El ahorro puede proceder de la distribución de la información en linea, de la facilidad para el establecimiento de la comunicación, de la sustitución de servidos o de la centralización de operaciones.

Si necesita distribuir información a una gran cantidad de usuarios, probablemente pueda realizar la misma tarea a través de un sitio Web de manera más económica. Independientementede que se trate de suministrar listas de precios, catálogos, procedimientos documentados, especificaciones lfflücas o cualquier otra información, es probable que resulte más barato colocar la información en la Web que imprimirla y distribuirla en papel, en especial si la información cambia de manera regular. El resultado es el mismo, ya se trate de distribuir ofertas y responder a ellas rápidamente o de facilitar la comunicación directa de cliente con el mayo~ta o el fabricante, eliminando intermediarios: los precios bajarán o los ingresos aecenln.

La sustitudón de servicios que supongan un gasto por su versión electrónica puede reducir los costes. Un buen ejemplo es el de Egghead.com. Los responsables de esta empresa decidieron cerrar sus almacenes de ordenadores y centrarse en actividades de comercio electrónico. Aunque la construcción de un sitio de comercio electrónico de cierta entidad cuesta dinero, el coste de mantener una cadena de más de 10 tiendas es mucho mayor. La sustitución de un servido existente tiene sus riesgos. Entre ellos, la pérdida de clientes que no utilicen Internet.

El cambio de Egghead.com no funcionó. La empresa cerró su$ tiendas ffsicas durante el boom de las empresas .comen 1998 y se declaró en bancá'Wotæn 2001.

La centralización tambim puede contribuir a reducir costes. El Dlintenimiento de una gran cantidad de sedes físicas requiere el pago de rentas y gastos indirectos, el personal que trabaja en ellas y los costes del mantenimiento de inventario en cada sede. Un negocio en Internet puede situarse en un único punto y resultar accesible a todo el mundo.

Riesgos y amenazas

Todos los negocios deben hacer frente a riesgos, competidores, robos, cambios en las piefe.endas del pliblico y a desastres naturales, entre otros. La lista es interminable. Sin embargo, muchos de los riesgos a los que deben hacer frente las compaftías de comercio electrónico son muy inferiores o resultan irrelevantes con respecto a otras iniciativas. Estos riesgos incluyen:

- Piratas informáticos.
- Fracaso en la atracción de suficiente negocio.
- Fallos de hardware informático.
- Fallos de alimentación, de comunicacióny de redes.
- Fiabilidad de los servicios de distribución.
- Competencia excesiva.
- Errores de software.

EL_HOLISTICO@HOTMAIL.COM - 2006

- Cambios en las polfticas e impuestos gubernamentales.
- Límites de la capacidad del sistema.

Piratas informáticos

Las amenazas mú conocidas para los sitios de comercio electrónico proceden de los usuarios de ordenadores malintencionados conocidos como piratas infonMticos. Todos los negocios corren el riesgo de convertirse en el objeti\·o de los criminales, pero no hay duda de que el comercio electrónico atrae la at~ción de los piratas infonMticos con una variedad de intenciones y habilidades.

Los piratas pueden atacar por el desafio o la fama que supone sabótear un sitio, para robar o para obtener artículos y servicios gratuitos.

La protección de un sitio implica una combinación de tareas:

- Mantener copias de seguridad de la información importante.
- Aplicar polfticas de contratación que atraigan a personal honesto y manterÍer su fidelidad (los ataques más peligrosos proceden del interior).
- Adoptar precauciones basadas en software, como seleccionar software seguro o actualizarlo.
- Formar al personal para identificar objetivos y puntos débiles.
- Auditar y registrar las actividades para detectar las intrusiones o los intentos de intrusión.

Los ataques más efectivos sobre sistemas informáticos aprovechan puntos d&iles conocidos, como contrasellas fáciles de averiguar, malas configuraciones y versiones antiguas de software. La adopción de unas cuantas medidas puede desanimar a los atacantes con poca experiencia y garantizar la disporúbilidad de una copia de segdridad si ocurre lo peor.

Fracaso en la atracción de suficiente negocio

Aunque los ataques de los piratas infonMticos son muy temidos, la mayor parte de los fracasos en el ámbito del comercio electrónico viene detennu,ada por factores económicos tradicionales. La creación y comercialización de un sitio de comercio electrónico de gran tamaAo exige fuertes inversiones. Las compai\fas aceptan la pérdida de dinero a corto plazo si ello conlleva un refuerzo de la marca en el mercado y un incremento de dientes e ingresos futuros.

La caída de las empresas .com provocó el declive de numerosas empresas debído a la falta de capital para mantener la pérdida del merado. En la larga lista de &acasos —dos destaca la empresa europea boo. ca., que agotó sus fondos y fue absorbida tras invertir mú 120 millones de dólares en seis meses. No es que Boo no vendiera, sino que gastaba mucho mú de lo que obtenía.

Fallos de hardware informático

No hace falta decir que si su negocio depende de un sitio Web y falla una parte esencial de uno de los equipos, el resultado se verá afectado.

En los sitios Web con mucho tráfico o cuyo mantenimiento activo resulte esencial está justificada la existencia de sistemas redundantes para que el fallo de uno afecte al funcionamiento de todo el sistema. Como con todas las amenazas, debe determinar si la posibilidad de desactivar su sitio Web por un día a la espera de que se realicen las reparaciones pertinentes justifica los gastos en equipo redundante.

Resulte bastante sencillo configurar varios equipos en los que se ejecute Apache, MySQL y PHP, y gracias a la replicación de MySQL, también se pueden sincronizar con facilidad pero esta solución implica un aumento del hardware, de la infraestructura de red y de los costes de alojamiento.

Fallos de alimentación, comunicación, redes y distribución

Internet se basa en una compleja red de proveedores de servicios. Si su conexión con el resto del mundo falla, no podrá hacer mucho más que esperar a que el proveedor vuelva a instalar el servicio. Otro tanto se puede decir con respecto a las @rta del fluido el#ctrlco a 111 b1le.lga1 y a Qlro Cipe de paree relaeienacloe con las compailías de reparto.

Existe una solución que depende del presupuesto y consiste en contratar varios servicios a proveedores distintos. El coste se incrementará pero .si uno de los proveedores falla, podrá recurrir a otro. Los cortes breves de suministro e1Ktñco se pueden superar invirtiendo en generadores.

Competencia excesiva

Si va a abrir una tienda en una calle, no le costará mucho hacer un estudio exacto sobre la competencia con la que se va a enfrentar. Sus competidores serán priJlcipalmente negocios que vendan artículos similares y que estén situados en los alrededores. Además surgirán otros. En el ámbito del comercio electrónico, el terreno es mú incierto.

Los competidores dependerán de los costes de envío y pueden encontrane en cualquier parte del mundo. Además, estarán sujetos a las fluctuaciones de las divisas y a los costes de la mano de obra. En Internet la competencia es encarnizada y cambia a gran velocidad. Si trabaja en una categoría popular, pueden aparecer competidores todos los días.

No hay _muchoque pueda hacer para eliminar el riesgo de la competencia, pero st se mantiene al tanto de los nuevos desarrollos, puede estar seguro de que su iniciativa resultar4 competitiva.

EL_HOLISTICO@HOTMAIL.COM - 2006

Errores de software

Los negocios basados en software resultan vulnerables a los errores de las aplicaciones informátic:.,s. Para reducir la probabilidad de errores vitales, seleccione. software que resulte fiable, uigne tiempo suficiente para probarlo tras cambiar partes de su sistema, disponga de un proceso de prueba formal y no permita que se realicen cambios en el sistema activo sin probarlos antes en otra parte. Para redudr la gravedad de los resultados, realice copiu de seguridad actualizadas de todos los datos, mantenga configuradones operativas de software al realizar un cambio y supervise el funcionamiento del sistema para ~ proble111fl rápidamente.

Cambios en las políticas e impuestos gubernamentales

La legisladdn sobre las empresa basadas en Internet varia según los países. Las posibmdades ydan desde que no exista legislación alguna a que se encuentre en fase de desarroll~ .que ~tf sufidentemente desarrollada. Esta situadón es poco probable que continue asi. Como resultado, algunos modelos empresariales podrían ver- ~ amenazados, regulados o e~dos por la legisladón futura. Ademú, surgirm 1D1p~estos. No hay forma de evitar estu cuestiones. La única solución consiste en estar atento a la nueva legislación y mantener el sitio dentro de la ley. Considere la posibilidad de integrarse en un grupo de presión cuando vayan surgiendo estas cuestiones.

Límites de-la capacidad del sistema

Una cuestión que debería tener presente al diseilar su sistema es el crecimiento. Lo normal es que su sistema reciba cada vez mú tráfico por lo que deberia diseilarse de forma se pueda escalar para hacer frente a la nueva demanda.

Si el aedmiento es limitado puede incrementar la capacidad del sistema adquiriendo hardware mú rápido. Pero los equipos tienen un limite en cuanto a la velocidad. ¿Está escrito su software para que al alcanzar ese punto, se puedan separar sus partes para compartir la carga entre varios sistemas? ¿Puede su base de datos procesar variu petidones simultmeas procedentes de diferentes equipos?

Hay muy pocos sistemas que puedan hacer frente a un crecimiento ingente si esfuerzos, pero si diseila su sistema teniendo en cuenta e1 factor de la escalabilidad, podrá identificar y eliminar los cuellos de botella cuando crezca su base de clientes.

Seleccionar una estrategia

Algunos aeen que Internet cambia demasiado nlpido como para poder realizar planes que resulten eficaces. Pero esa mutabilidad convierte en fundamental la

labor de planifiaición. Si no establece metas y selecciona una estrategia, se venl obligado a improvisar ante los cambios en lugar de anticipane a ellos.

Tras examinar algunos de los objetivos dpicm de un sitio Web comercial y algunas de las amenazas principales a las que hacer frente, es probable que ya tenga pensada alguna estrategia para el suyo.

Esta estrategia debe identificar un modelo de negocios. En general, el modelo de negocios suele ser una iniciativa cuyo funcionamiento ya se ha probado, pero en ocuiones puede tratarse de una idea en la que tenga fe. ¿Adaptará su modelo de negocios actual a la Web, imitará a un competidor o aead un servicio pionero?

A continuación

En el siguiente capftulo, abordaremos el tema de la seguridad en el comercio electrónico y le ofrel:eremos una descripción general de los tfrminos, amenazas y tt!c:nicas de seguridad.

EL_HOLISTICO@HOTMAIL.COM - 2006

13

Aspectos de seguridad relacionados con el comercio electrónico

En este capítulo se analiza el papel de la seguridad en el comercio electrónico. Veremos quiln puede estar interesado en nuestra información y cómo podrían obtenerla, los principios implicados en la creación de una política para evitar este tipo de problemas y algunas tecnologías disponibles para salvaguardar la seguridad de un sitio Web incluida la criptografía, la autenticación y el rastreo.

En este capítulo se examinarán los siguientes aspectos:

- Importancia de la información
- Amenazas contra la seguridad
- Crear una política de seguridad
- Equilibrio entre usabilidad, rendimiento, coste y seguridad
- Principios de autenticación
- Utilizar autenticación
- Fundamentos de la criptografía
- Criptogralfa de clave pública
- Firmas digitales
- Certificados digitales
- Servidores Web seguros
- Auditorias y registros

- Cortafuegos
- Copia de seguridad de datos
- Seguridad física

Importancia de la información

Al analizar el tema de la seguridad, lo primero que tenemos que valorar es la importancia de los datos que estamos protegiendo, tanto parat nosotros como para los posibles atacantes.

Puede resultar tentador creer que siempre deberia aplicarse el nivel mú alto seguridad a todos los sitios, pero la protección tiene un coste. Antes de decidir el esfuerzo o los gutos que se desean invertir en el sistema de seguridad, es necesario determinar el valor de la información.

El valor de la información almacenada en el equipo de un usuario que utilice su ordenador para divertirse, el de una empresa, el de un banco y el de una organización militar es diferente, al igual que varían los esfuerzos que invertirá un atacante para obtener acceso a dicha información. Tenemos que determinar el valor que tienen los contenidos almacenados en nuestros equipos para un visitante malintenciónado.

Los usuarios que utilizan el ordenador como pasatiempo dispondrán de tiempo limitado para profundizar en el tema de la seguridad de sus sistemas o en aumentarla. Como la información que almacenann en sus equipos es poco probable que interese a otros usuarios, los ataques serúl poco &~entes y el ~erzo invertido en ellos será limitado. Sin embargo, todos los usuarios de equipos con conexión a la red deberían tomar una serie de precauciones ya que incluso los equipos con los datos menos interesantes pueden utilizarse como plataformas para lanzar ataques a otros sistemu.

Los equipos militares son un objetivo obvio tanto para individuos como para gobiernos extranjeros. Para atacar a los gobiernos es necesario disponer de muchos medios, por lo que resulta aconsejable invertir en personal y otro tipo de recursos para garantizar la aplicación de todu las precauciones posibles.

Si es responsable de un sitio de comercio electl'Ónico, su atractivo para los piratu informáticos se situará entre estos dos extremos, por lo que los recursos y los esfuel'Z()S que dedique al respecto tambim se encontrarm entre dichos extremos.

Amenazas contra la seguridad

¿Cuáles son los riesgos que se demen sobre su sitio? ¿Qu6 amenazas existen? En un capítulo anterior se analizaron algunas de las amenazas a las que se expone el comercio electrónico. Muchas de ellas estm relacionadas con la seguridad.

Lu amenazas variarán en función del tipo de sitio Web pero se pueden dtar las siguientes:

- Exposición de datos confidenciales
- P&dida o destrucción de datos
- Modificar datos
- Denegación de servicio
- Errores en el software
- Repudio

En las siguientes secciones se analiza cada una de ellu.

Exposición de datos confidenciales

La información almacenada en los equipos o la transmitida de un equipo a otro, puede ser confidencial. Podria tratarse de información destinada Úl'.\iljlmente a determinadas penonas, como listu de precios de mayoristas, informa,ión suministrada por un cliente, como su contrasei\a, datos de contacto o el n1lmero de una tarjeta de crédito.

No deberla guardar la información q~ no desee compartir en suaervidor Web. Los servidores Web no son el lugar mú aconsejable en el que guard!r información secreta. Si almacena los datos sobre su nómina o su plan para la dominación del mundo en un ordenador, procure que no sea el mismo que el utilizado como servidor Web. Los servidores Web son equipos diseftados para su acceso público por lo que sólo deberían contener información que se ofrezca públicamente o que se acabe de recoger del público. Para reducir el riesgo de la exposición de datos, debe limitar los métodos de acceso a la información y lcis usuarios que disponen de dicho acceso, lo cual implica desarrollar las labores de diseño teniendo presente la seguridad, configurar el servidor y el software de manera correcta, programar con atencióq, realizar pruebas exhaustivas, eliminar los servicios innecesarios del servidor Web y exigir autenticación.

Dise1\e, configure, codifique y pruebe atentamente su sitio Web para reducir los riesgos de un ataque y, lo que es más importante, para reducir la probabilidaC:..d, que un error exponga accidentalmente la información.

Suprima los servicios innecesarios de su servidor Web para reducir el número de puntos d6biles potenciales. Cada servicio que est6 ejecutando puede resultar vulnerable. Es necesario mantener todos los servicios actualizados para eliminar los puntos vulnerables conocidos. Los servicios no utilizados pueden resultar peligrosos. Si no utiliza el comando rcp, ¿porqu6 tenerlo instalado? Si indic;a al instalador que su equipo es un host de red, las distribuciones más .importantes"de Linux y Windows NT instalarúl una gran cantidad de servicios que no n~ta y que debería eliminar.

EL_HOLISTICO@HOTMAIL.COM - 2006

La autenticación implica solicitar a los usuarios que demuestren su identidad. Cuando un sistema sabe quién realiza una petición, puede decidir si dicha persona dispone de acceso. Existen varios métodos de autenticación de los cuales sólo se utilizan dos de forma generalizada: las contrasellas y las firmas digitales. Los analizaremos más adelante.

CD Universe ofrece un buen ejemplo del coste, tanto en dinero como en reputación, que supone permitir la exposición de información confidencial. A finales de 1999, un pirata informático, que se hacía llamar Maxus, se puso en contacto con CD Universe para indicarles que disponía de 300.000 números de tarjetas de crédito que había sustraído su sitio Web. Por la destrucción de esta información pedió una recompensa que CD Universe se negó a pagar. Como resultaao, CD Universe ac.abó en las primeras páginas de todos los periódicos de mayor tirada cuando Maxus comenzó a hacer públicos los números de lu tarjetas de crédito de sus usuarios.

Los datos también corren el riesgo de exposición en sus trayectos por la red. Lu redes TCP/IP incorporan una gran cantidad de funciones que han contribuido a convertirlo en el estmtdar de facto para la conexión de diversas redes como Internet. Sin embargo, la seguridad no se incluye entre ellu. Los protocolos TCP/fP dividen los datos en paquetes y los reenvían de equipo en equipo hasta que alcanzan su destino. Por lo tanto, los datos pasarán a través de una gran cantidad de ordenadores hasta llegar a su destinatario, como se ilustra en la figura 15.1, y cualquiera de ellas puede ver los datos en su tránsito.

Flguni 15.1. Los datoll enviados a través de lnlilmet rec:omin una gran cantidad de equipos poCenc:ialmentelnMDUl'OS.

Para ver la ruta que siguen los datos desde un equipo hasta el equipo de destino, puede utilizar el comando traceroute (en un equipo UNIX). Este comando devuelve las direcciones de los equipos a través de los cuales han pasado los datos para alcanzar el host de destino. Si los datos van dirigidos a un equipo de su propio país, podrían atravesar una decena de equipos düerentes. Si fueran destinados a un equipo situado fuera del país, el número de intermediarios podría superar la veintena. Si su organización es de gran tamailo y consta de una red compleja, los

datos podrían recorrer huta cinco equipos antes de abandonar el edificio. Para proteger la información confidencial, puede cifrar los datos antes de enviarlos a través d~ una red y descifrarlos en el otro extremo. Los servidores Web suelen utilizar el sistema SSL (Secure Soc:ket Layer), desarrollado por Netscape para rulizar esta tarea mientras los datos navegan entre los servidores Web y los navegadores. Se trata de una forma bastante barata en términos de costes y de esfuerzos de proteger lu transmisiones, pero como el servidor necesita cifrar y descifrar los datos en lugar de enviarlos y recibirlos simplemente, el número de visitantes por segundo al que puede dar servicio el equipo se reduce drásticamente.

Pérdida o destrucción de datos

La pérdida de datós puede resultar más costosa que su apropiación indebida. Si lleva meses construyendo su sitio, recogiendo información sobre usuarios y pedidos, ¿sería capaz de determinar el coste, en tiempo, reputación y dinero, resultante de perder toda esa información? Si no ha realizado ninguna copia de seguridad de esos datos, se verá obligado a volver a escribir el sitio Web y a comenzar desde cero.

Un pirata informático podría colarse en su sistema y formatur su disco duro. También podría ocurrir que un programador o un administrador poco atento eliminen el disco sin querer o que un disco se estropee. Los discos duros giran miles de veces por minuto y en ocasiones fallan. Según la ley de Murphy, fallará el disco más importante y mucho después de realizar una capia de segundad

Puede adoptar varias medidas para reducir la probabilidad de perder datos. Proteja sus servidores contra ataques informáticos. Reduzca al mínimo los empludos con acceso a su equipo. Contrate únicamente personal competente y atento. Compre discos duros de calidad. Utilice la arquitectura RAID para permitir el uso de varios discos duros como un disco más rápido fiable.

Independientemente del causante de las pérdidas de datos, sólo hay una forma de protección real: lu copias de seguridad. La realización de copias de seguridad no es exactamente una tarea divertida y con suerte no necesitará recurrir a ellas, pero su función es vital. Asegúrese de volcar sus datos de manera periódica y de probar el procedimiento para asegurarse de que resultan recuperables. Asegúrese de almacenar las copias de seguridad lejos de los equipos habituales. Aunque es poco probable que los locales de su empresa se quemen o sufran alguna otra catástrofe, el almacenamiento de una copia de seguridad fuera de ellos es una medida de seguridad barata y eficaz.

Modificar datos

Aunque la pérdida de datos puede causar muchos dailos, su modificación puede resultar aún peor. ¿Qué ocurriría si alguien logra acceder a sus datos-,. tos modifica? La eliminación completa de los datos no tardaría mucho tiempo en detectarse, pero pregúntese cuánto tiempo llevarla detectar su modif1eaci6n.

EL_HOLISTICO@HOTMAIL.COM - 2006

Entre las formas de modificación se incluyen cambios en los archivos de datos o ejecutables. La intención de un pirata infonn4tico puede ser modificar el aspecto de la Web u obtener beneficios fraudulentos. Mediante la sustitución de archivos ejecutables con versiones saboteadas, un pirata infonn4tico que haya logrado entrar una vez en el sistema podría crear uíia puerta trasera secreta para futuras visitas.

Puede proteger los datos contra su modificación al viajar por la red mediante una firma. Esta Nc:nica no impide la modificación de los datos pero, si el receptor comprueba que la firma coincide al recibir los archivos, sabr6 que el archivo no se ha modificado. Si cifra los datos para protegerloscontra la visualizaciónno autorizada, resultaré muy diffciles de modificar durante el trayecto sin su detección.

La protección de los archivos almacenados en un servidor frente a su modificación exige el uso de las funciones de permisos de acceso a archivos que incorpore su sistema operativo y protege el sistema de accesos no autorizados. La aplicación de permisos de archivos permite autorizar el uso del sistema, pero no modificar los archivos del sistenta ni de otros usuarios. La falta de un sistema de permisos es una de las razones que convert(an a los sistemas operativos Windows 95, 98 y ME en poco aptos para su uso como servidores.

Detectar las modificaciones puede resultar dificil. Si descubrimos que alguien há penetrado en nuestro sistema de seguridad, ¿cómo podremos saber si se han modificado archivos importantes? Algunos archivos, como los archivos de datos que se almacenan en una base de datos, estm ideados para cambiar con el paso del tiempo. Sin embargo, la intención de otros es permanecer invariables desde su iñ;stalación a menos que se actualicen deliberadamente. La modificaciónde programas y datos puede resultar engaftosa, y aunque los programas puedan reinstalarse si se sospecha que han sido modificados, puede que no sepa quf versión de los datos est6 •intacta".

Existen aplicaciones que permiten comprobar la integridad de los archivos, como Tripwire. Esta aplicación registra información sobre archivos en un estado seguro, probablemente inmediatamente después de la instalatjón, y permite utilizarla en un momentoposterior para verificar si los archivos han variado. Puede descargar la versión comercial o gratuita de http://www. tripw'tre. com.

Negación de servicio

Una de las amenazas más difíciles de evitar es lo que se denominado Negación de servido (DoS, en ingl&). Se produce cuando alguien impide a los usuarios acceder a un servid!) o retrasa el acceso a un servicio en el que el tiempo es crucial.

A principios de 2000, se produjo una oleada de ataques DoS distribuidos a importantes sitios Web. Entre las vfctimas se encontraban Yahoo!, eBay, Amazon, E-Trade y Buy.com. Estos sitios est6n acostumbrados a niveles de tr6fico que nosotros sólo podemos concebir en sueftos, pero siguen siendo vulnerables a este tipo de ataques. Aunque los atacantes no ganan nada con conseguir colapsar un sitio, el propietario del mismo puede perder ingresos, tiempo y reputación.

Algunos sitios esperan con~ntrar sus transaccionesen determinadas horas. Algunos sitios de apuestas en línea experimentan una mayor demanda antes de un evento deportivo importante. Una de las formas en que los atacantes pretendían ap~!._echarsede~a~q~ ~ en 2004 era exigiendo una cantidad a los sitios de apuestas o les atacarían durante las horas de mayor demanda.

Una de las razones de la dificultad de evitar este tipo de ataques es que se pueden producir de ~fenmws (onnas. Se puede instalar un programa en el equipo de la víctima que utilice el tiempo del próc:esador del sistema, el envío inverso de correo basura o recurrir a una herramienta automatizada. El correo basura inverso envía c:orreo basura en el que se incluye el destinatario como emisor. De esta forma, el destinatario recibirá miles de respuestas de usuarios enfadados.

Existen herramientasautomatizadas que pueden lanzar ataques DoS a un destino concreto. No hay que ser un genio para buscar los puntosciS)iles de una serie de equipos e instalar la herramienta en los mismos. Como el proceso est6 automatizado, un atacante puede instalar la herramienta en un solo host en cuestión de segundos. Una vez afectados varios equipos, reciben instrucciones para inundar el destino con tr6fico de red.

Protegerse de ataques DoS es una dificil tarea. Si realiza una ——ueda, puede ~brir los puertos predeterminados que utilizan las herramienl!lDoS habituales y cerrarlos. Puede que su enrutador cuente con un mecanismir., para limitar el porcentaje de tráfico que utiliza determinados protocolos a)IIIO por"ejemplo ICMP. Resulta más sencillo detectar los host de una red utilizados para atacar a otros que proteger nuestros equipos de posibles ataques. Si todos los administradores de redes se comprometieran a vigilar su propia red, los ataques DoS no serian un problema. Como existen tantos métodos de ataque, la única defensa realmente eficaz consiste en controlar el comportamiento del tnfíc:o normal y contar con una serie de expertos para que adopten las medidas oportunas en caso de que se produzca alguna situación fuera de lo normal.

Errores en el software

Es posible que el software que haya comprado, obtenido o escrito incluya errores graves. Dados los cortos plazos de desarrollo asignados a los proyectos Web, es muy probable que el software incluya errores. Todas las inidativas comerciales basadas en procesos inform6ticos resultan vulnerables a software con errores.

Los errores en el software pueden causar todo tipo de comportamientos impredecibles incluida la indisponibilidad del servido, lagunas de seguridad, pérdidas financieras y servidos deficientes. Entre las causas habituales de los errores se pueden citar las malas especificadones técnicas, sup,osidones erróneas realizadas por los desarrolladores y pruebas incompletas.

Malas especific1tdone~ técnicas

Cuanto más escasa y ambigua resulte su documentación de diseño, más probable es que el producto final incluya errores. Aunque puede que resulte obvio esped-

EL_HOLISTICO@HOTMAIL.COM - 2006

ficar la anulación de un pedido si se rechaza la tarjeta de crédito de un cliente, puedo hablarles de un sitio con un alto presupuesto que indufa este error. Cuanta menos experiencia tengan los desarrolladores con el tipo de sistemas con el que están trabajando, más precisas deberían ser las especificaciones técnicas.

Suposiciones erróneas hechas por los desarrolladores

Los diseftadores y los programadores de un sistema necesitan realizar una gran cantidad de suposiciones. Es de esperar que las documenten y que resulten acertadas. Sin embargo, las suposiciones pueden resultar erróneas. Por ejemplo, un desarrollador podría asumir que los datos de entrada serán rilidos, que no Incluinin caracteres inusuales o que la cantidad de caracteres introducidos será inferior a un tamai\o dado. También podría asumir que no tendrán lugar dos acciones simultáneas que entren en conflicto o que una tarea de procesamiento compleja lleve más tiempo que una tarea sencilla.

Suposiciones como éstas se pueden deslizar porque suelen ser ciertas. Un pirata informático puede aprovecharse de un desbordamiento de blifer generado por una suposición hecha por un programador con respecto a la longitud máxima de los datos de entrada o un usuario legítimo puede obtener mensajes de error confusos o abandonar la visita porque a los desarrolladores no se les ocurrió prever que el nombre de una persona pueda incluir un apóstrofe. Este tipo de errores se puede detecrilr y solucionar mediante una combinación de pruebas y revisión detallada ddJ:6diga...

Históricamente, el sistema operativo o las debilidades del nivel de aplicación explotldas por los piratas informática se relacionan con los desbordamientos de búfer o con los errores de sincronización.

Pruebas incompletas

Resulta pnicticamente imposible verificar todas las posibles entradas de los usuarios en todos dispositivos de hardware existentes, con todos los sistemas operativos posibles y utilizando todos los parámetros de usuario disponibles. Esto es especialmente cierto en el caso de los sistemas basados en la Web.

Es necesario implementar un plan de pruebas bien diseñado que verifique todas las funciones del software en un conjunto representativo de equipos. Un conjunto de pruebas bien planeado debería probar todas las lineas de código del proyecto al meno.suna vez. De manera ideal, estas pruebas deberían automatizarse para poder ejecutarlas en los equipos seleccionados con Ínuy poco esfuerzo.

El problema más importante de esta operación es que se trata de un trabajo repetitivo muy poco atractivo. Aunque hay gente a la que le gusta romper cosas, hay muy pocas a las que le guste romper la misma cosa una y otra vez. Es importante que en este proceso participen otras personas además de los desarrolladores originales. Uno de los objetivos principales de las pruebas es detectar las suposiciones erróneas realizadas por los desarrolladores. Es probable que una persona no implicada realice suposiciones diferentes. Además, los profesionales no suelen mostrarse muy inclinados a buscar errores dentro de su trabajo.

Repudio

El último riesgo que considerar es el repudio. El repudio tiene lugar cuando una parte implicada en una transacción niega haber tomado parte en ella. En el comercio electrónico puede tratarse de una persona que realice un pedido a través del sitio Web y que rechace la autorización para realizar el cargo sobre su tarjeta de crédito, o una persona que acepte algo procedente del correo electrónico y que después afirme que alguien falsificó dicho c:oneo.

De manera ideal, las transacciones financieras deberían garantizar la seguridad de que ninguna de las partes las repudian. Ninguna parte podría denegar su partí-apaaoñ en una transacción o, para ser más exacto, ambas partes podrían demostrar de manera terminante las acciones realizadas por la otra parte ante un tercero, como por ejemplo un tribunal. En la práctica, esto no suele ocurrir. La autenticación ohece ciertas garantías sobre la parte con la que se está tratando. Los certificados digitales de autenticación emitidos por una organización de confianza brindan una gran fiabilidad. También deberían poder certificarse los contenidos de lo mensajes enviados por cualquiera de las partes. No sirve de mucho poder demostrar que Corp Pty Ltd nos ha enviado un mensaje si no podemos probar que hemos recibido exactamente los que los han enviado. Como se indicó anteriormente, la firma o el cifrado de los mensajes dificulta su alteración de manera subrepticia.

Pan las transacciones entre las partes con una relación ya establecida, los certi-6cados di&itale& yel YS& de eemlfflicaciones cifl1das o fil macias propotdonaii una forma efectiva de limitar el repudio. Si las transacciones son aisladas, como el contacto inicial entre un sitio de comercio electrónico y un usuario que utilice una tarjeta de cM<lito, esta opción no resulta tan pnictica.

Las compai\fas de comercio electrónico deben estar dispuestas a probar su identidad y a contratar los servicios de una autoridad de certificación como VeriSign (http://www. verisign. com) o Thawte (http://www. thawte. com) para garantizar la autenticidad de la compañía. ¿Estará dispuesta esa misma compai\ía a rechazar a todos los clientes que no estén dispuestos a adoptar la misma medida para demostrar su identidad? En las pequeñas transacciones, los comerciantes están dispuestos a admitir un determinado nivel de &aude o riesgo de repudio a cambio de mantener el nivel de transacciones.

Equilibrio entre usabilidad, rendimiento, coste y seguridad

Por su propia naturaleza, la Web es un lugar peligroso. Este medio ~ diseAado para permitir que numerosos usuarios soliciten servicios desde sus equipos. La mayor parte de estas peticiones realizan peticiones de páginas Web perfectamente legítimas, pero al conectar sus equipos a Internet también existe la posibilidad. de que la gente realice otros tipos de conexiones.

EL_HOLISTICO@HOTMAIL.COM - 2006

Aunque resulta tentador asumir que es apropiado establecer el nivel de seguridad m4s alto, no suele ser así. Si desea estar completamente protegido, tendrá que colocar todos los equipos en una caja fuerte sin conexión a la red. Para que los equipos resulten accesibles y se puedan utilizar, es necesario rebajar .el nivel de seguridad de alguna manera.

Debemos buscar un equilibrio entre la seguridad, la usabilidad, el coste y el rendimiento. Si aumentamos la seguridad de un servido podemos reducir su ~apa~ cidad de uso, por ejemplo, al limitar lo que puede hacer la gente o pedir que se identifiquen.

El incremento de la seguridad también puede reducir el nivel de rendimiento de sus equipos. La ejecuci6n de software para lograr que el sistema resulte m'5 seguro (como sistemas de encriptación y detección de intrusos, escáneres de virus y operaciones de registro de usuarios) absorbe recursos. Se necesita mucho m4s potencial de procesamiento para suministrar una sesión cifrada, como una conexión SSL a un sitio Web, que una sesión normal. La pérdida de rendimiento se puede compensar con la adquisición de equipos o hardware más rápido que estf especialmente dise-i'lado para operaciones de encriptaci6n.

El rendimiento, la usabilidad, el coste y la seguridad se pueden considerar como objetivos contrapuestos. Deberá examinar los pros y los contras de cada opción y tomar una decisión que logre un compromiso en función del valor de la información, del presupuesto, de la cantidad de visitas esperadas y de los obstáculos que considere que aceptarán los usuarios legítimos.

Crear una política de seguridad

Una politica de seguridad es un documento que describe

- La filosofía general sobre seguridad en su organización
- Qué se debe proteger: software, hardware y datps
- Quién es responsable de la protección de estos elementos
- Estándares de seguridad e indicadores para medir el cumplimiento de dichos estándares

A la hora de desarrollar la politica de seguridad se pueden aplicar las mismas reglas utilizadas para escribir un conjunto de requisitos de software. En esta política no debería hablarse de implementaciones o soluciones, sino de requisitos de seguridad y de objetivos dentro del entorno. Y no debería actualizarse a menudo.

Debería crear un documento en el que se establezcan las directrices utilizadas para medir el cumplimiento de la política de seguridad en un entorno dado. Puede establecer directrices diferentes para las distintas partes de la organización. Seria como un documento de diseño o un procedimiento manual en el que se recoja qué se está haciendo para garantizar el nivel de seguridad requerido.

Principios de autenticación

El objetivo de la autenticación es demostrar que alguien es quién dice ser. Existen muchas formas de suministrar autenticación pero como en el caso de muchas medidas de seguridad, cuanto mú seguros son los métodos más problem4ticos resallan de utilizar.

la IKnicas de autenticaciónincluyen el uso de contraseflas, firmas digitales, medidas biométricas mediante escáneres de huellas y medidas que implican el uso de hardware como tarjetas inteligentes. En la Web s6lo se utilizan dos de manera extendida: las contrasellas y las firmas digitales.

Las medidas biométricas y la mayor parte de las soluciones de hardware implican el uso de dispositivos de entrada especiales, lo que limitaría el acceso a aquellos usuarios autorizados con ordenadores equipados. Esta opción podría resultar aceptable o incluso deseable para obtener acceso a los sistemas internos de una organizaci6n, pero reduce gran parte de las ventajas de permitir el acceso al sistema en la Web.

Las contraseñas resultan fáciles de implelllentar, sencillas de utilizar y no requien!n el uso de dispositivos de entrada especiales. Suministr~ determinado nivel de autenticación pero es posible que resulten insuficientes..-ra sistemas de alta seguridad.

El concepto de una contraseña es sencillo. El usuario y ~· sistema conocen la contrasei'la. Si otro visitante afirma ser un usuario y conoce su contrasei'la, el sistema lo aceptará. Este sistema resulta seguro siempre y cuando nadie más sepa o pueda adivinar la contrasei'la. Las contrasei\as presentan en sí mismas varios puntos débiles y no suministran autenticación segura.

Muchas contraseflas resultan fkiles de averiguar. Si permite que los usuarios seleccionen sus propias contraseflas, un *S0* por cien~ escogerá contrasei\as fáciles de descubrir, como vocablos de un diccionario o el nombre de usuario. Puede obligar a los usuarios a incluir números o signos de puntuación en sus contrasei'las, aunque esta medida afectará a la usabilidad.

Otra opción consiste en mostrar a los usuarios cómo seleccionar mejores contraseflas pero de todas maneras el 25 por dento de ellos seguirá utilizando contrasei\as sencillas de adivinar. Puede implementar políticas de contraseflas que impidan a los usuarios seleccionar combinaciones fáciles comparando las contraseflas introducidas con un diccionario o exigiendo el uso de números o signos de puntuacióno una combinación de letras en mayúsculas y minl1sculas. El peligro de una polftica de este tipo es que los usuarios legftimos no logren recordar las contrasei\as escogidas. Las contrasellas difíciles de recordar aumentan la probabilidad de que los usuarios realicen una acción poco segura como escribirlas en una nota y pegarla en sus monitores. Es necesario educar a los usuarios para que no escriban sus contraseflas ni las comuniquen por teléfono a alguien que afirme estar trabajando en el sistema.

Las contraseflass6lo deberían capturarse electrónicamente. Meditante el uso de programas para capturar pulsaciones del teclado en una terminal o la información

EL_HOLISTICO@HOTMAIL.COM - 2006

que circula por la red, los piratas informáticos pueden capturar pares de nombres de usuario y contrueftas. Puede limitar las oportunidades de capturar contraseñas cifrando el tráfico de red.

A pesar de sus defectos potenciales, las contraseñas son una forma sencilla y relativamente efectiva de autenticar usuarios. Puede que el nivel de anonimato no resulte adecuado para la seguridad nacional, pero son una solución perfecta para comprobar el estado del pedido de un cliente.

Utilizar autenticación

La mayor parte de los navegadores y servidores Web incorporan mecanismos de autenticación. Los servidores Web se pueden configurar para solicitar un nombre de usuario y una contraseña para permitir el acceso a los archivos de determinados directorios del servidor.

Cuando se necesite introducir un número de usuario y una contraseña, el navegador presentará un cuadro de diálogo parecido al de la figura 15.2.

Figura 15.2. Los navegadores Web solicitan a los usuarios que se autentiquen al intentar visitar un directorio de acceso restringido en un servidor Web.

Tanto Apache como el servidor 115 de Microsoft permiten proteger de manera sencilla una parte del sitio o el sitio en su totalidad de esta forma. PHP y MySQL incorporan muchas otras formas de lograr el mismo objetivo. MySQL resulta más rápido que la autenticación integrada. PHP suministra una autenticación más flexible o presenta la solicitud de forma más atractiva.

En un capítulo posterior veremos algunos ejemplos de autenticación.

Fundamentos de la encriptación

Un algoritmo de encriptación es un proceso matemático que transforma información en una cadena aparentemente aleatoria de datos.

Los datos de los que se parte son texto sin procesar, aunque no es importante para el proceso lo que represente la información (que se trate de texto o de otro tipo de datos). La información ci&ada se conoce como texto encriptado, aunque se parezca poco a un texto en la mayor parte de los casos. La figura 15.3 muestra el proceso de encriptado en forma de diagrama de flujo. El texto que se desea cifrar se introduce en un motor de encriptadón, que podría ser un dispositivo mednico, como los equipos Enigma de la segunda guerra mundial, auque en la actualidad presenta prácticamente siempre el aspecto de un programa informático. El motor devuelve el texto encriptado.

Figura 15.3. La •ncrlplación toma texto sin procesar y lo transforma en texto encriptado aparentemente aleatorio.

Para crear el directorio protegido cuya solicitud de autenticación se muestra en la figura 15.2, utilizamos el tipo de autenticación más básica de Apache. (En el siguiente capítulo le mostraremos cómo utilizarlo.) Este método cifra las contraseñas antes de almacenarlas. Creamos un usuario con lá contrasefta paaa vord. &ta se cifró y se almacenó como aWDuA3X3H. mc2. Como puede observar, el texto de. partida y la cadena cifrada no se parecen en nada.

Este sistema da Qó:ado no ea rewenible, MuGhaæonlH&ei\aæse almaeenan utilizando un algoritmo de encriptación de una sola dirección. Para poder determinar si una contraseña introducida resulta correcta, no es necesario descifrar la contraseña almacenada. En su lugar, bastará con cifrar la contraseña introdutj.da y cqmpararla con la versión almacenada.

Muchos procesos de encriptación se pueden invertir, aunque no todos. El proceso inverso al encriptación se conoce como descifrar. La figura 15.4 muestra un proceso de encriptación de doble sentido.

Figura 15.4. El proceso de encrlptaclórtDma texto sin procesar y lo transforma en texto cifrado con aspecto aleatorio. El proceso de descifrado toma el texto encrlptado y lo tranafonna en el texto de partida.

La criptografía tiene unos 4000 ai\os de antigüedad, pero alcanzó la mayoría de edad en la segunda guerra mundial. Su crecimiento desde entonces ha seguido de cerca al desarrollo de las redes de ordenadores. En un principio Sólo era utilizada

EL_HOLISTICO@HOTMAIL.COM - 2006

por las institucionesmilitares *y* organizaciones financieps. En los aftos 70 se extendió su uso *y* en los aftos 90 se hizo omnipresente. En los últimos aftos, la encriptación ha pasado de ser un concepto que la gente sólo conocía por las pelfculas de espfas *y* de la segunda guerra mundial a conyertirsE en algo-de lo que se habla en los periódicos *y* se utiliza cada vez que se compra algo a trav& de navegadores Web.

Existen muchos algoritmos de encriptación disponibles. Algunos utilizan una clave secreta o privada,*y* otros, utirwn una clave pdbllca *y* una-claveprivada distinta.

Encriptacién de clave privada

La encriptaciónde clave privada se basa en usuarios autorizadosque conocen o disponen de acceso a una clave. Esta clave debe mantenerse en secreto ya que de lo contrario los mensajes cifrados podrian ser leídos por personas no autorizadas. Como se muestra en la figura 15.4, tanto el remitente (la persona que cifra el mensaje) como el receptor (el que descifra el mensaje) tienen la misma clave.

El algoritmo de clave secreta mú utilizado es DES(del ingl& Data Encryptation Standard, Estmdar de encriptación de datos). Fue desarrollado por IBMen los años 70 *y* fue adoptado como el es~ar americano para comunicaciones comerciales *y* gubernamentalesno clasificadas. La velocidad de los ordenadores ha aumentado mucho desde entonces *y* DES ha quedado obsoleto desde al menos 1998.

Otros sistemas de claves secretas conocidos son RC2, RC4, RC5, triple DES e IDEA. El sistema triple DES resulta bastanteseguro (paradógicamente, este sistema el5 dos veces mú seguro que DES; si necesita un sistema tres veces mú seguro, puede escribir un programa que implementeun algoritmo DES quíntuple). Utiliza el mismo algoritmoque DES, pero se aplica tres veces con hasta tres claves diferentes. El mensaje de texto se cifra con la primera clave, se descifra con la segunda*y* se vuelve a cifrar con la tercera.

Uno de los defectos de la encriptación con clave secreta es que para enviar un mensaje seguro a alguien, es necesario disponer de un forma segura de poderles enviar la clave secreta. Si dispone de una forma sefll'a de enviar una clave, ¿por qui no utilizarla para enviar el mensaje?

Afortunadamente,en 1976 se produjo un avance significativo cuando Diffie *y* Hellman publicaron el primer mitodo de clave pública.

Encriptación de clave pública

La encriptación de clave pública se basa en dos claves, una clave pública *y* una clave privada. Como se ilustra en la figura 15.5, la clave pública se utiliza para cifrar mensajes *y* la clave privada para descifrarlos.

La ventaja de este sistema es que la clave pdblica, como su nombre indica, se puede distribuir públicamente. Todo el mundo al que le entreguemos nuestra clave pública puede enviamos un mensaje seguro. Mientras mantengamos en secreto nuestra clave privada, sólo nosotros podremos descifrar el mensaje.

Figura 15.5. La 811Ci lptaci6,i de cIllve públlc:e ullllza clavel dlsllntas para cifrar y clelc:lfrar rmenujes.

El algoritmo de clave pdblica má utilizado es RSA. desarrollado por Rivest, Shamir *y* Adelman en Mrr *y* publicado en 1978.RSA en un sistema propietario pero la patente venció en septiembre de 2000.

La capacidad para transmitir una clave pública de forma abierta *y* sin necesidad de preocuparse por que resulte visible para una tercera parte supone una gran venta¡a. Un sistema de clave pública se utiliza para transmitir la clave para un sistema de clave secreta que se utilizar4 ldurante el resto de una CC)lllunicación de sesión. Esta complejidad ailadida resulta tolerable porque los sistemas de clave secreta son unas 1000veces má ralpidos que los sistemas de clave¡,piblica.

Firmas digitales

Las firmas digitales estm relacionadas con la criptograffa de clave pública, pero invierten los papeles de las claves públicas *y* privadas. Un remileñlé puede cifrar *y* firmar digitalmente un mensaje con su clave secreta. Cuando se recibe el mensaje, el receptor puede descifrarlo con la clave secreta del remitente. Como el remitente es la única persona con acceso a la clave secreta, el receptor puede estar bastante seguro de la procedencia del mensaje *y* de que no se ha alterado.

Las firmas digitales resultan muy útiles. Garantizan la procedencia de los mensajes *y* dificultan el rechazo del envio por parte del remitente. De todos modos, es importante tener en cuenta que aunque el mensaje se ha cifrado, cualquier persona que tenga la clave pdblica puede leerlo. Si bien se utilizan las mismas ticnicas *y* claves, la finalidad del uso de la encriptación en este caso es impedir la modificación de los mensajes *y* su repudio, no su lectura.

Como el cifrado de clave pública resulta bastante lento en el caso de mensajes de gran tamafto, se suele utilizar otro tipo de algoritmo, llamado función hash, para me;orar la eficacia. La función hash calcula un mensaje implfcito o valor hash pan cualquier mensaje que se le indique. El valor generado por el algoritmo no es importante. Lo importante es que el resultado sea fijo, es decir, que sea el mismo cada vez que se utiliza una entrada dada, que sea pequei\o*y* que el algoritmo sea r41pido.

Las funciones hash mú habituales son MDS*y* SHA. Una función hash genera un mensaje implícito que coindde con un mensaje dado. Si disponemos de ambos mensajes, podremos verificar si el texto se ha variado, siempre *y* cuando estemos

EL_HOLISTICO@HOTMAIL.COM - 2006

seguros de que no se ha tocado el mensaje implícito. La forma más habitual de crear una firma digital consiste en crear un mensaje implfcito para todo el mensaje utilizando una función hash rápida y, a t"Ontinuadón, cifrar únicamente el mensaje implícito utilizando una algoritmo de clave pública. Seguidamente se puede enviar la firma con el mensaje a través de cualquier método normal no seguro.

Cuando se recibe un mensaje, se puede realizar su comprobación. La firma se desc:ifruá utilizando la clave pública del remitente. Se genera un valor hash para el mensaje utilizando el mismo método utilizado por el remitente. Si el valor hash descifrado coincide con el valor hash generado, podemos estu seguros de que el mensaje procede del remitente y no se ha alterado.

Certificados digitales

La posibilidad de \"erificar que un mensaje no se ha alterado y que procede de un usuuio o de un equipo dado es un gran avance. Para las interacciones comerciales, resultaría incluso mejor disponer de la posibilidad de vincular a dicho usuario o sewidor a una entidad legal real como una persona o compaiUa.

Un certificado digital combina una clave pública con los detalles de una organización o individuo en un formato digital firmado. En un certificado, nosotros tenemos la clave pública de la otra parte, por si queremos enviar un mensaje cifrado y teud1enws los detalles ele la Offll parte, que podelftOSestar segmos ele que no se han alterado.

"El problema en este caso es que la fiabilidad de la información depende de lo fiable que sea el remitente ya que todo el mundo puede generar y firmar un certificado utilizando otra identidad. En las transacciones comerciales, resulta útil utilizar una tercera parte de confianza que acredite la identidad de los participantes y los detalles de sus certificados. Estas terceras partes se conocen como CA (del inglés Certifying Authorities, Autoridades de certificación). Las autoridades de certificación expiden certificados digitales a individuos y compaiüas sujetas a una serie de comprobaciones de identidad. Las dos CA más conocidas son VeriSign (ht tp: / /www. verisign. ccm)y Thawte (ht tp: / /www. thawte. cOIII), pero existen otras. VeriSign y Thawte pertenecen a la misma compailia y hay muy poca diferencia entre ambas. Algunas menos conocidas, como Equifax Secure (www. equifaxsecure. com), resultan bastante más baratas.

Las autoridades firman un certificado para acreditar que se ha comprobado la identidad de una peISOnao compailfa. Tenga en cuenta que el certificado no es una refelencia ni una declaración de solvencia. No garantiza que se está tratando con una entidad fiable. Lo único que garantiza es que en caso de que nos engallen, es pl'ONble que ob~gamos una direcdón fisica y una entidad o persona contra la que dirigir la demanda.

l.GS certificados suministran una estIUctura de confianza. Si confiamos en la CA, podemos confiar en aquellas parta en las que confía la CA y a su vez en aquellas parles en las que confía la parte certificada.

Los certificados digitales se suelen utilizar para t"Onferirun aire de respetabili· dad a un sitio Web de comercio electrónico. Con un certificado emitido por una CA conocida, los navegadores Web pueden establecer conexiones SSL a nuestIO sitio sin generar cuadros de diálogo de advertencia. Los servidores Web que permite conexiones SSL se suelen llamar servidores Web seguros.

Servidores Web seguros

Puede utilizar el servidor Web Apache, el servidor 115 de Microsoft o cualquier otro servidor Web comercial para establecer comunicaciones seguras con navegadores a través de SSL. Apache permite utilizar un sistema operativo de tipo UNIX, que suelen resultar más fiables pero más diffciles de configurar que 115. El servidor Apache se puede utilizar en una plataforma Windows.

El uso de SSL sobre 115 implica sencillamente instalar 115, generar un par de claves e instalar el certificado. El uso de SSL en Apache requiere la instalación de tres paquetes diferentes: Apache, Mod_SSLy OpenSSL. También puede obtener lo mejor de los dos mundos comprando Stronghold. Stronghold es un producto comercial disponible en http: / / stronghold. redhat. com por unos 1000 dólares. Está basado en Apache pero tiene forma de archivo binuio autoinstalable y lleva pret"Onfiguradoel sistema SSL. De esta forma, se obtiene la fiabilidad de UNIX así como un pioducto d.e Udl iNtaladén oon uiatenc:ia ÑGN,a d.el pl'O'l•eede"

Las instrucciones de instalación para los dos servidores Web más populares, Apache e 115, se incluyen en el apéndice A. Puede comenzar a utilizu SSL de forma inmediata si genera sus propios certificados digitales pero los visitantes que acudan a su sitio recibirán una advertencia de sus riavegadoies indicándoles que hemos firmado nuestros propios certificados. Para poder utilizar SSL con eficacia, necesitará un certificado publicado por una autoridad de certificación.

El proceso exacto para lograrlo varia según la autoridad de certificación, pero por regla general deberá demostrar que su empresa está reconocida legalmente, que consta de una dirección física y que es el titular del nombre de dominio pertinente.

Necesitará generar una solicitud de firma de certificado. El proceso para logrulo varia de un servidor a otro. Encontrará las instrucciones en los sitios Web de las autoridades de certificación. Stronghold e 115 incorporan un proceso basado en cuadros de diálogo mientras que Apache requiere la introducción de comandos. Sin embargo, el proceso es esencialmente el mismo para todos los servidores. El resultado final es una solicitud de firma de certificado (CSR)cifnda. Las CSR presentan un aspecto parecido al siguiente:

```
--·-·BSGIR RBlr CBRTIPICATI RBQUEST·-··-
MIIBuwIBAAJCllgQCLnlXXlfllllllbtsStp9wY&BVTPulU9bpllmbrbCvg    ..l}'tdTe&11S
84p711Gepq5CQjfOL4Hjda+912x1to8ux8kCII098Xg9qlCCY4511Zk+q5GyOOLISOD     '
lcQBwbloUP&SaSTzOll~PB1pI3bHxfO&aYallfY1iDiFl¡,1BrUdua+pE4SOIVAPLB
SV9PSlll7IIIOg121r5Rl2oQ01AoGAWSPllyfVXPAPlb2GDb+cf9'7k44vtlll+RxpelG
```

EL_HOLISTICO@HOTMAIL.COM - 2006

9blf8a9L31SII0III0JllfDL1ny7clStYV91VTVllek1dYuaa89}'BkPnly711CU1111111Snlt
4UjtPDkllbX9j5YbCRGLallel,5AT541DlU3102/dlalLo'lf9FP1r1jlly99HJ4LRY919
HkXVaewC9Y81181'112Qfltll C&Jlltll3ab+&eHQ4Deo11tx1&27llJl511eQLwkPGa+WtYSZ
jG5tw4tqqovmJ+IP2F/5G&PI2DQP7QDvXllaAUljXeuijuWo27S2ebbQtl9ZRTZVO
jGnl9BCO•Ill9HQllkJ7va35mclSkk3Vllq3ebwllGC«r1voeiv2JIX2IQIVII0'l'Rp7ap
la7Q1XaJtwl87XXb!NltPO
·····IIID 11111 CIRTJFJCATS lllQUBST·····

Tras adquirir la CSR, pagar la cuota correspondiente, conseguir la documentación que prueba nuestra existencia y verificar que el nombre de dominio que estamos utilizando es el mismo que el incluido en la documentación corporativa, estari en disposición de firmar un certificado con una autoridad de certificación.

Cuando la autoridad de certificados emita el suyo, deberi almacenarlo en su sistema e indicarle a su servidor Web dónde encontrarlo. El certificado final es un archivo de texto parecido al CSR mostrado anteriormente.

Auditorias y registros

Su sistema operativo le permitiri registrar todo tipo de eventos. Los eventos en los que puede estar interesado desde el punto de vista de la seguridad incluyen errores de red, acceso a archivos de datos concretos como los de configuración o el registro de NT y llamadas a programas como au (que se utiliza para convertirse en otro usuario, por regla general el usuario raíz, en un sistema UNIX).

Los archivos de registro pueden servir de ayuda para detectar comportamientos erróneos o malintencionados cuando tienen lugar. También se pueden utilizar para determinar cómo tuvo lugar un problema o una intrusión si se examina tras observar problemas. Los archivos de registro presentan dos problemas principales: el tamai'lo y la veracidad.

Si establece los criterios para detectar y registrar problemas en el nivel de mayor paranoia, terminad obteniendo registros gigantescos que resultan muy diffciles de examinar. Para trabajar con archivos de registro de Slan tamafto, tendri que recurrir a una herrunienta existente o derivar algunas secuencias de comandos de la política de seguridad establecida para realizar búsquedas de eventos "interesantes" dentro de los registros.

El proceso de auditar los registros puede realizarse en tiempo real o de manera periódica.

Los archivos de registros resultan vulnerables a ataques. Si un intruso dispone de acceso de usuario raíz o de administrador a nuestro sistema, podd alterar los archivos de registros para borrar sus huellas. Unix permite registrar eventos en un equipo diferente, lo que implica que el intruso deberi acceder a dos equipos al menos para cubrir sus huellas. Windows dispone de funcionalidad similar, pero no resulta tan sencilla de implementar.

El administrador del sistema puede realizar auditorias periódicas, pero ademú, podríamos contratar un servicio de auditorias externo para que controlara el comportamiento de los administradores.

Cortafuegos

m oll¡etivo de los cortafuegos es alejar una red del mundo exterior. De la misma forma que los c:orlafuegos de un edificio o de un coche impiden que el fuego se extienda a otros compartimentos, los cortafuegos de red impiden que el caos se extienda por una red.

Los cortafuegos estm diseñados para proteger los equipos de una red del mundo exterior. Filtran y rechazan el trffico que no cumple las reglas establecidas en ellos. Restringe las actividades de la gente y de los equipo situados fuera del co~¿nes, los cortafuegos también se pueden utilizar para restringir las actividades de los equipos incluidos en su interior. Un cortafuegos puede limitar los protocolos de red que se pueden utilizar, los host a los que se pueden conectar o forzar el uso de un servidor proxy para reducir costes en t4nninos de ancho de banda.

Un contrafuegos puede ser un dispositivo de hardware, como un enrutador con reglas de filtrado o un programa de software que se ejecute en un equipo. En cualquier caso, el cortafuegos necesita interfaces para las dos redes ~ conjunto de reglas. Su misión consiste en supervisar todo el tdfico que intentapasar de una red a otra. Si el trifico cumple las reglas establecidas, sed dirigido a otra red; de lo contrario, se detendd o se rechazad.

Los paquetes se pueden filtrar por tipo, dirección de origen, dirección de destino o puerto. En algunos casos, los paquetes pueden rechazarse simplemente y en otros pueden desencadenar entradas en el registro o acbvar alarmas.

Copia de seguridad de los dates

No se p~ede subestimar la impo~da de las copias de seguridad en ningún plan de recuperación de desastres. El hardware y los inmuebles se puede asegurar Y sustituir, o los sitios alojados en cualquier host, pero si su software Web personal desaparece, ninguna entidad aseguradora podd sustituirlo.

Debe hacer copias de seguridad periódicas de todos los componentes de su si~o Web (piginas est4ticas, secuencias de comandos y bues de datos). La frecuenaa depende del dinamismo del sitio. Si se trata de un sitio completamente estitico, bastad con hacer una copia cuando se modifique. Sin embargo, el tipo de sitios que estamos tratando en este libro es probable que cambien con frecuencia, en especial si se reciben pedidos en linea. La mayor parte de los sitios con un tamafto razonable deben alojarsé en un servidor con sistema RAID (del ingl& Redundant Amy of Inexpensive Disks, Matriz redundante de discos independientes) que admite fundones de réplica. De esta forma, queda cubierta la probabilidad de que ocurra un fallo en el disco duro. Considere, sin embargo, qué ocurriría en una situación que afecta a la matriz, al equipo o al edificio entero.

EL_HOLISTICO@HOTMAIL.COM - 2006

La frecuencia con la que realizar los volcados de seguridad debería corresponderse con el volumen de actualización. Estas copias de seguridad deberían almacenarse en un soporte diferente y a ser posible en un lugar distinto y seguro contra incendios, robos o desastres naturales.

Existen muchas aplicaciones para realizar funciones de copia de seguridad y recuperaciones. En nuestro caso, nos centraremos en cómo volcar un sitio construido con PHP y equipado de una base de datos MySQL.

Copia de seguridad de archivos generales

Los volcados de archivos HTML. PHP, imágenes y otros archivos que no sean de base de datos se puede realizar con facilidad en la mayor parte de los sistemas utilizando software para la realización de copias de seguridad.

Entre las herramientas gratuitas más utilizadas se puede mencionar AMANDA desarrollada por la University of Maryland. Esta utilidad se incorpora en la mayor parte de las distribuciones de UNIX y también se puede utilizar para realizar copias de seguridad en equipos Windows a través de SAMBA. Si desea saber más sobre AMANDA, diríjase a http://www. amanda. org.

Cppia de seguridad y restauración de bases de datos MySQL

El proceso de volcado de seguridad de una base de datos en funcionamiento resulta más complicado ya que no queremos copiar ningún dato mientras la base de datos se está modificando.

En un capítulo anterior encontrará instrucciones sobre cómo volcar y restaurar una base de datos MySQL.

Seguridad física

Las amenazas de seguridad consideradas hasta el momento hacen referencia a elementos intangibles como el software, pero no debería descuidar la seguridad física de su sistema. Debe prenr un sistema de aire acondicionado y sistemas de protecdón contra el fuego, personas (torpes o criminales), fallos en el summistro eléctric:oy fallos en la red.

Su sistema debería estar protegido bajo llave en un lugar seguro. El lugar concreto dependerá de cada caso pero puede ser una habitación, una jaula o un armario. S6lo sed~ conceder permiso de acceso a esta habitación al personal pertinente. El personal no autorizado podría, voluntaria o involuntariamente, desconectar los cables o intentar eludir los mecanismos de seguridad utilizando un disco de ieinicio.

Los rodadores de agua pueden causar tanto daño a los componentes electrónicos como el fuego. En el pasado, se utilizaron sistemas de extinción de incendios

alimentados por gas halón para evitar este problema pero en la actualidad la producción de este gas está prohibida por el Protocolo de Montreal sobre sustancias que reducen la capa de omno, por lo que los sistemas para la extinción de incendios deben utilizar otras alternativas menos damnas como el g¡as argón o el dióxido de carbono. Puedeleer más al respecto en http://www.epa.gov/Ozone/anap/fire/qa.html.

En determinados lugares, se producen bieves cortes de fluido electrónico de manera habitual. En sitios con climatologías extremas o cables enterrados se suelen producir cortes de fluido de larga duración. Si el funcionamiento continuo del sistema es un factor importante debería invertir en un S!J (Sistema de alimentación ininterrumpido). El coste de un SAi capaz de mantener el fluido el4ctrico para un equipo durante 10 minutos asciende a unos 300 dólares. Si se necesita un sistema para cubrir cortes de suministro más largos o más equipos, el coste-se elévará. Los cortes de suministro de larga duración requieren un generador que alimente el sistema de re&igeradón y los equipos.

Al igual que los fallos de suministro e14ctrico, los fallos de red, que pueden durar minutos o horas, quedan fuera de nuestro control y se producen i.rreme~iablemente de tiempo en tiempo. Si el mantenimiento de la red es vital, es aconsejable contratar los servicios de varios proveedores de servidos. El coste será superior, pero en caso de fallo, su sitio seguirá siendo visible, aunque se vea afectada su capacidad.

Este ,: de problemas podrían llevarle a considerar la opción de ubicar sus equipos ntro de una infraestructura dedicada. En un negoao de tamafto inedio puede que no resulte justificable el uso de un SAi que mantenga el ~uido elfctrico durante varios minutos, varias conexiones de red redundantes y sistemas antiincendios. Sin embargo, en una infraestructura de calidad en la que se alojen dentos de equipos similares sí resultaría justificable.

A continuación

En el siguiente capítulo, nos centraremos de manera específica en la autenticación, función que permite a los usuarios acreditar su identidad. Examinaremos varios mftodos diferentes, incluido el uso de PHP y MySQL para autenticar a nuestros visitantes.

EL_HOLISTICO@HOTMAIL.COM - 2006

164 4. Manipular cadenas y expresiones regulares

"

En este capítulo veremos cómo implementar varias técnicas de PHP y MySQL para autenticar a un usuario.

Entre los aspectos que analizaremos se incluyen los siguientes:

- Identificar visitantes
- Implementar el control de acceso
- Autenticación básica
- Utilizar la autenticación básica de PHP
- Utilizar la autenticación básica . htaccess de Apache
- Utilizar la autenticación básica con 115
- Utilizar la autenticación con mod_auth_mysql
- Crear procesos de autenticación personalizados

Identificar visitantes

La Web es un medio relativamente anónimo, pero a menudo resulta útil saber quién está visitando nuestro sitio. Afortunadamente para la privacidad de los visitantes, los datos que se pueden obtener sin su colaboración son bastante escasos.

EL_HOLISTICO@HOTMAIL.COM - 2006

Con un poco de trabajo, los servidores pueden averiguar bastantes datos sobre los equipos y las redes que se conectan a ellos. Los navegadores suelen identificarse, para lo cual indican al servidor qué navegador, versión y sistema operativo se está utilizando. Se puede determinar qué resolución y profundidad de color está utilizando el visitante en su pantalla asf como el tamaño uignado a la ventana del navegador Web.

Cada equipo que se conecta a Internet tiene una dirección IP exclusiva. Esta dirección permite deducir algunos datos sobre el visitante. Se puede determinar a quién pertenece la IP y, aproximadamente, la ubicación geográfica del visitante. Algunas direcciones resultan más útiles que otras. Por regla general, los usuarios con conexiones permanentes a Internet tendrán una direcdón permanente. A los dientes que llamen a un proveedor de acceso a Internet se les suelen asignar direcciones para su uso temporal.

Si vuelve a ver dicha dirección es probable que esté siendo utilizada por un equipo distinto y si vuelve a encontrar al visitante es probable que esté utilizando una dirección IP diferente. Las direcciones IP no resultan tan útiles para identificar a usuarios como puede parecer en un principio.

Por suerte para los usuarios Web, ninguno de los datos que ofrece el navegador sirve para identificarles. Si desea saber el nombre u otra información sobre un visitante, tendrá que preguntarles.

Muchos sitios Web utilizan razones convincentes para obtener datos de los usuarios. El New York Times (http: //www.nytimes.com) ofrece su contenido gratis pero--sólo- a- aquellas pel'90MS que- estén dispuestas a stHninistffr eetes- EOIftO su nombre, sexo e ingresos. El sitio Slashdot (http://www. slashdot. org) permite a los usuarios registrados participar en los debates con un apodo y personalizar la interfaz que ven.

La mayor parte de los sitios de comercio electrónico registra información sobre sus usuarios cuando realizan el primer pedido. De esta forma no tendrán que volver a hacerlo en los pedidos siguientes.

Tras solicitar y recabar información de los visitantes, necesitamos una forma de asociar dicha información con sus usuarios correspondientes cuando vuelvan a visitamos. Si partimos del supuesto de que cada persona visitará su sitio desde una éuenta determinada y desde un equipo dado, y que cada visitante utiliza únicamente un equipo, podemos almacenar una cookie en el equipo del usuario para identificarle. Sin embargo, no ocurre siempre es así. Con frecuencia, la gente comparte un equipo y muchos usuarios utilizan varios equipos para navegar. Por lo tanto, no quedará más remedio que preguntar al visitante quién es de nuevo. Además de solicitar que se identifique, tendrá que pedirle alguna prueba que confirme que es quién dice ser;

Como se indicó en un capítulo anterior, el proceso de pedir a los usuarios que demuestren su ic;lentidad se conoce como autenticación. El método habitual de autenticación utilizado en los sitios Web consiste en pedir a los usuarios que introduzcan un nombre de usuario y una contraseña. El proceso de autenticación se suele utilizar para conceder o rechazar el acceso a págínas o recursos concretos, pero puede resultar adecuado o útil para otros fines como la personalización.

Implementar el control de acceso

Un control de acceso sencillo no resulta diffcil de implementar. El código incluido en el listado 16.1 d,vuelve tres posibles resultadoe. Si el archivo se carga sin parámetros, mostrará un formulario HMTL solicitando un nombre de usuario y una contrasefta. En la figura 16.1 se muestra este tipo de formulario.

RflIN-16,1.-Este fonnulerio tffMI-picle-t los- visitantes que introdtlZCIln un-nombre de usuario y una contrasefta para poder ac:ceder.

Si se han introducido los parámetros pero no son correctos, se mostrará un mensaje de error. En la figura 16.2 se muestra dicho-mensaje.

Figura 16.2. Cuando los usuarios introducen información incorracta, tenemos que devolver un ~je de error. En un sitio real pueda que le intarase proporcionar un mensaje más agradable.

EL_HOLISTICO@HOTMAIL.COM - 2006

Si los parámetros se han introducido y son correctos, se mostrará el contenido secreto, como se ilustra en la figura 16.3.

Figura 11.3. Cuando se introducenlos detalles correctoa, la secuencia de comandos muestra el contenido.

El código para crear la funcionalidac:filustrada en las figuras 16.1, 16.2 y 16.3 se muestra en el listado 16.1.

Uatado 11.1. secrelphp. Código PHP y HTML para proporcionanun sencillomecanismo de autenticación.

```
c?php
   //cree nombre• corto• para la• variable•
• Sna.. • S POST(•na ..•J;
• Spaaavord-· S_JOST('pa••-rd'I ı

   if(etipty(Sna ..lifel11pty|Spa••-rd11
   {
      //11 vi•itante tiene que introducir un nombre y una contra•efla

      <hl>Plea•e Log Inc/hl>
      Thia page i• aecret.
      cform ..thod•°poat• action•°•ecret.php°>
      <table border•°l">
      ctr>
         cth> U•erna.. c/th>
         ctd> <input type••texc• n...••name°> c/td>
      c/tr>
      <tr>
         cth> P•••vord c/th>
         <td> cinput type•°pa•avord• — · ·p•••vord°> c/td>
      </tr>
      ctr>
         ctd colepan.•2• align••center°>
            <input type•••ubiaJ.t• value•°Log In°>
```



```
         c/td>
      c/tr>
      e/table>
      e/fon>
c?php
   )
   el•• if(S--••'uaer•••Spa••vord••'ila••'I
   {
      // el D0llbre ~ la cobtraaek del viaitante eon correcto•
      echo '<hl>Bere it ialc/hl>'1
      echo •I bet you are glad you can ... thia ••cret paga.•;

   el••
   {
      // el D0llbre y la contra•efla del vi•itante no •0n correctoe
      echo •cll1>0o A—y!c/hl>' r
      echo 'You are not autboriaed to viev thi• reeource.'1

?>
```

El c6digo del listado 16.1 constituye un sencillo mecanismo de autenticación para conceder IICCt!IO a una página a los usuarios autorizados, pero presenta algunos problemas importantes. Esta secuencia de comandos

- Incluye un nombre de usuario y una contraseila en su interior
- Almacena la contraseña en forma de texto sin procesar
- Sólo protege una página
- Transmite la contrasefta en forma de texto sin procesar

Estos problemas se pueden resolver con diferentes grados de esfuerzo y éxito.

Almacenar contraseñas

Existen muchos lugares en los que almacenar nombres de usuario y contraseñas mejores que dentro de una secuencia de comandos. Los datos introducidos dentro de una secuencia de comandos resultan diffciles de modificar. Se podrfa escribir una secuencia de comandos que se modificara a sí misma pero no es una buena idea ya que supondría tener una secuencia de comandos en el servidor, que se ejecutaría en él pero que ~ podrían modificar o esaibir sobre ella. Si almacenamos los datos en otro archivo dentro del servidor resultad ms sencillo escribir un programa para agregar y eliminar usuarios y alterar contraseñas.

El uso de secuencias de comandos u otros archivos de datos con funciones de almacenamiento presenta un límite en cuanto al número de usuarios que se pueden incluir sin que se vea afectada la velocidad de la secuencia de comandos. Si tiene previsto almacenar y realizar búsquedas en un archivo con una gran cantidad de elementos, debería considerar la opción de utilizar una base de datos, como se comentó anteriormente. Como regla, si el número de elementos que necesita almacenar y buscar supera los 100, utilice una base de datos en lugar de un archivo.

EL_HOLISTICO@HOTMAIL.COM - 2006

El u&o de una base de datos para almacenar nombres de usuario y contrueilas compliad la secuencia de comandos, pero a cambio permitid autenticar a muchos usuarios rápidamente. Tambiin permitirá escribir con facilidad una secuencia de comandos para agregar nuevos usuarios, para eliminar usuarios y para permitir que los usuarios cambien sus contraseffas.

En el listado 16.2 se incluye una secuencia de comandos para autenticar visitantes utilizando una base de datos.

LI..«to 11.2. secretc:lb.php. Utilizar MySQL para mejorar el sencillo mecanismo de auteollcaci6n.

```
clpbp
   $Dalle•   S_POST[•—•J1
   SPA••-rd-•   S_P08T1'PA•••ord'J;

   iflliee et1S_POST1'iuume'J1••tiee•t0S_fOST1'pa••word•J]!
   (
      //El vi•itaate ti•n• que introducir un aC1lbr• y una coat~a••1la

      c111..11-     Log lac/hb
   4  Tbi• page i•  .. crét.
      cform —tJlacl ··poet• actioa••••cretdb.pbp•>
      ctable bordar••1•>
      ctr>
         cth> Oe•raa-  c/th>
         ctd> cinput type~•t..t•      .........> c/td>
      c/tr>
      Ctr>
         cth> Pa•eword c/th>
         ~td> d.aput t~'!~JMeeword·  aa-··pail.-rd•> c/td>
      c/tr>
      ctr>
         ctd col•paa•"2" aliga••c•nt•r">
            ciaput type•"•ubait• valuee•Log ID">
         c/td>
      c/tr>
      c/tabl•>
      c/form>
c?pbp
   )
   elN
   (
      // Htableaca la coaexi6a a my..l
      Smy1lql • -,.qli_c —— ct( 'localbo•t•.  •wbaut!I•,  'flhauth' 1
      U (1 s.yeq11
      (
       • ec:bo '⊂—t c-ect to dataN ... •;
         mlit;
      )
      /.( Hlecc.ioae la ha•e de datoe pertinente
      ,..1ected • myaqli_•elect_dbl S-,,.ql, •auth' l,
      if(ISmyaql)
      (
```



```
         ec:tlo •caanot ••l•ct datatla•e.•;
         •xit;


      // coaeult• la bu•  d• dato• para dateniaar  •i exi•t•  un regi•tro
      // que coincida
      Squery • ···l•ct couatC•t from autbariaild ueer• wllere
            na-  •  •S —— ' aad                    -
            pae• • 'Spa•-rd••,

      $renlt  • 111Y9qli_qu•ryC $,ayeql, $query 11
      if( ISre•ult1
      (
         echo 'C:-t  rua qu•ry. '1
         exit,
      }
      $-row • -,.qli_f•tch_row( Sresult 11
      $count • $row(0)1

      if ( $couat > 0 1
      (
         // •l II01111re y la contruda   dal vi•itante -   correctoe
         ec:bo •clU>Ben it ie le /111>'1
         ecbo 'I b•t you ar• glad you can ...  thi• •ecret page.•1
      }
      •]••
      (
         // •l mmbr• y la contruda   del vieitaate ao -   correcto•

         echo '1'ou are aot autboriaed to view thi• rneaarce.•,
```

La base de datos q~ estamos utilizando se puede crear conectándose a MySQL como usuario raíz de MySQL y ejecutando los contenidos del listado 16.3.

Ustaclo 11.3. creatNuthdb.sql. Estas c:onsultas MySQL crean la base de datos auCh, la tabla auth y dos usuarios de ejemplo.

```
create databa•• auth;
u•• auth1
creat• table auchoriaed_ue•re   -    varcharC20I,
                                paaawozd vazchar(t0I,
                                   priMry  UF  1-)
                              1;
iaeert iato authoriaed_uaer• valu••   •uae.rnam,
                                      'pa•_..  );
ia•ert into autb.oriaed_uer• valu••   'ceacuer•,
                                      ehall 'pa•-rd')   1 i
grant •elect oa auth.•
            to  •weJ,auth'
            ideatified by  'W8haUth' /
fll18h p~ivil999a,
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Cifrar contraseñas

Independientemente de dónde almacenemos las contraseftas (en una base de datos o en un archivo), no conviene hacerlo como texto sin procesar porque corremos un riesgo innecesario. Podemos utilizar un algoritmo de comprobación aleatoria (hash) de una din!cción para obtener una mayor seguridad con muy poco esfuerzo.

PHP cuenta con una serie de funciones de hash unidin!ccionales. La m4s antigua *y* menos segura es el algoritmo Unix Crypt, que ofrece la función crypt ( ). El algoritmo Message Digest S (MDS), implementado en la funcion mdS (› es más robusto *y* est6 disponible en la mayoría de las versiones de PHP. Si no necesita compatibilidad con versiones anteriores de PHP, utilice la opción Secure Hash Algorithm 1 (SHA-1).

La funcion aba 1 ( ) de PHP proporciona un completo hash criptogr6fico unidireccional. Su sintaxis es la siguiente:

```
•tring •bal ( •tring •tr [, bool raw_output|I
```

Dada la cadena atr, la función devolver6 una cadena pseudoaleatoria de 40 caracteres. Si establece raw_output en true, obtendr6 una cadena de 20 caracteres de datos binarios. Por ejemplo, dada la cadena "paasword", ahal (› devuelve "Sbaa61e4c9b93f3f0682250b6cf8331b7ee68fd8". Esta cadena no se puede descifrar *y* convertir de nuevo en "paaaword"; ni siquiera por su creador, lo que puede que no resulte de gran utilidad a primera vista. Lo que hace que la función ahal () resulte útil es que su resultado es siempre el mismo. Por lo tanto, si utilizamos la misma cadena, la función ahal (› devolver6 el mismo resultado siempre que se ejecute.

En lugar de utilizar el siguiente código de PHP

```
if( $na.. •• 'u•erna ..' ••
      Spa••word •• •paaaword• 1

   //AK!eptado, la• contraaeftaa coinciden
```

podemos utilizar código como áte

```
if( Sna•• •• 'u•erna ..' ••
      •bal 1$paHwordl •• '5baanetc9b!13UfOU2250b6cHUlb7eeHfdl ' 1 1

   //Aceptado, la• contra••fta• coinciden
```

No necesitamos saber qué aspecto presentaba la contrasefta antes de aplicarle la función ahal (). Tan sólo necesitamos saber si la contrasefta escrita coincide como la cifrada inicialmente con la función shal ().

Comu se mencionó anteriormente, no conviene incluir los nombres de usuario *y* las contraseftas en una secuencia de comandos de manera expresa. Utilizaremos un archivo independiente o una base de datos para almacenar esta información.

Si estamos utilizando una base de datos MySQL para almacenar loa dalas de mtenticaci6n, podemos utilizar la función ahal (1 de PHP o la función SIIAl 1› de MySQL Estas funciones no generan el mismo resultado, pero estmt dildadas con elmismoftn.

... - Para utilizar SHAl( ), podriamos volver a escribir la consulta SQL del listado 16.2como

```
••lect count(•)  fr0II autborised_ua•r• wbere
  –    • '$UHr—· aacl
  pa•IIWOrd • •llal ( '$pa••word')
```

La consulta contar' el número de filas de la tabla authorized_uaera que tengan un nombie igual a los contenidos de $name *y* un valorpass igual al resultado producido por SHAl( ) aplicado a los contenidos de $password. Si los usuarios deben utilizar nombres de usuario exclusivos, el resultado de esta consulta sed o o 1. Recuerde que las funciones de hash suelen devolver datos con un tamafto fijo. En el caso de SHAl, son 40 caracteres cuando se representa como cadena. Asegweie de que las columnas de su base de datos tienen esa anchura.

Si examina de nuevo el listado 16.3, observar6 que hemos creado un usuario (•uaername •) con una contrasefta sin cifrar *y* otro usuario co~pua contruefta cifrada (•teatuaer •) para ilustrar estos dos enfoques.

Proteger páginas múltiples

La tarea de crear una secuencia de comandos como las de los listados 16.1 *y* 16.2 para proteger m6s de una p6gina resulta un poco mú complicada. "é:omo HTI'P no tiene estado, no existe un enlace o asociadón autom6tica entre las solicitudes consecutivas de un mismo usuario. Este hecho dificulta la tarea de mantener datos de una p6gina a otra, como la información de autenticación introducida por un usuario. La forma m4s sencilla de proteger varias p6ginas consiste en utilizar los mecanismos de control de acceso incorporados en su servidor Web, como veremos en breve.

Si queremos crear esta funcionalidad, podemos incluir partes de la secuencia de comandos ilustrada en el listado 16.1 dentro de las p6ginas que queremos proteger. Utilizando auto_prepend_file *y* auto_append_file, podemos adjuntar al prindpio o después el código requerido para cada archivo en directorios concretos. El uso de estos directorios se analizó en un capitulo anterior.

Si utilizamos este enfoque, ¿qué ocurrir' cuando nuestros visitantes recorran varias p6ginas de nuestro sitio? No podemos obligarles a volver a introducir sus nombres *y* contraseftas para cada p6gina que deseen ver. Podemos adjuntar los detalles introducidos a cada hipervfnculo de la página. Como los nombres de usuario pueden incluir espacios u otros caracteres no permitidos en los URL, deberfamos utilizar la función urlencode (› para codificar de forma segura estos caracteres.

Sin embargo, este enfoque presenta todavía algunos problemas. Como los datos se incluinn en p6ginas Web enviadas al usuario así como los URL que visiten, las

EL_HOLISTICO@HOTMAIL.COM - 2006

páginas protegidas que visiten resultarán visibles para todos los usuarios que utilicen el IIIÍSDIO equipo y retrocedan por las páginas almacenadas en caché o examinen el historial del navegador. La contrasefta se envía entre el navegador y el servidor cada vez que se solicita una página, lo cual no resulta muy recomendable.

Existen dos buenas formas de resolver estos problemas: la autenticación básica de HTTP y las sesiones. La autenticación básica resuelve el problema de las páginas almacenadas en caché, pero el navegador seguirá enviando la contraseila al servidor con cada petición. El control de sesiones resuelve ambos problemas. En primer lugar examinaremos el proceso de autenticación básica de HTl'P y dejaremos para capítulos posteriores el análisis del control de sesión.

Autenticación básica

La autenticación de usuarios es una tarea habitual, de ahí que HTl'P incorpore funciones que se ocupan de ello. Las secuencias de comandos o los servidores Web pueden solicitar autenticación desde un navegador Web. El navegador Web se encargará de mostrar un cuadro de diálogo o un elemento similar para obtener la información necesaria del usuario.

Aunque el servidor Web solicite nuevos de!alles de autenticación para ca~a petición del usuario, el navegador Web no necesitará pedir los detalles del usuano para'cad,~:J,~! ~ g~neralel _navegador.almaceJ!a esta informad~ mientras el usuario tenga una ventana del navegador abierta y la reenvía automáticamente al sen-idor Web cuando éste la soJicite sin interacción con el usuario.

Esta función de HTIP se denomina autenticación básica. Puede desencadenarla utilizando PHP o mecanismos incorporados al servidor Web. Examinaremos el método utilizado por PHP, por Apache y por 115.

El procéSOde autenticación básica transmite el nombre de usuario y la contraseila de un usuario en forma de texto sin procesar, por lo que no resulta muy seguro. HTIP 1.1 contiene un método algo más seguro conocido como autenticación de texto implícita, que utiliza un algoritmo de comprobación aleatoria (habitualmente MDS) para disfrazar los detalles de la transacción. Muchos servidores Web y las versiones más actuales de los mismos admiten la autenticación de texto implícita, pero no ocurre así con un número significativo de navegadores Web antiguos y una versión del estándar incluido en Microsoft IE e 115 incompatible con productos que no son de Microsoft.

Además del problema de compatibilidad con los navegadores Web, la autenticación de texto implícita resulta todavía poco segura. Tanto la autenticación básica como su variante por medio de resúmenes ofrecen un bajo nivel de seguridad. Ninguna garantiza a los usuarios que se está interactuando con el equipo deseado. Ambas permiten que un pirata informático pueda reproducir la misma petip~n en el servidor. Como la autenticación básica transmite la contrasefta del usuano en forma de texto sin procesar, cualquier pirata informático podría capturar paquetes y suplantar al usuario para realizar una petición.

La autenticación básica ofrece un nivel bajo de seguridad similar al de las conexiones a través de Telnet o FTP, ya que las contraseilas se transmiten en forma de texto sin procesar. La autenticación de texto implícita resulta un poco más segura, ya que cifra las contrasellas antes de transmitirlas. Para proteger todas las partes de una transacción Web de forma segura se puede utilizar SSL y los certificados digitales. Consulte el siguiente capítulo, si desea implementar un sistema de protección seguro. Sin embargo, en muchas ocasiones, resulta apropiado utilizar un sistema rápido aunque relativamente inseguro, como la autenticación básica.

La autenticación básica protege un dominio con nombre y exige el uso de un nombre de usuario y una contraseila válidos. Los dominios reciben nombres para poder alojar más de uno en el mismo servidor. Se puede dar el caso de que distintos archivos o directorios del mismo directorio formen parte de distintos dominios y que cada uno de ellos esté protegido por un conjunto diferente de nombres y contraseilas. Los dominios con nombre también permiten agrupar varios directorios en un host físico o virtual como dominio y utilizar una contraseila para protegerlos todos.

Utilizar autenticación básica en PHP

Por regla general, las secuencias de comandos son multiplataforma pero el uso de la autenticadón búica se basa en ~r.aRables de enteme estableeidas por el-aenidor. Para que una secuencia de comandos HTTPse pueda ejecutar en Apache utilizando PHP, en forma de módulo Apache o en 115 utilizando PHP é:omo módulo 15¡'\PI, es necesario detectar el tipo de servidor y comportarse de forma.lig'1'8111ente diferente. La secuencia de comandos del listado 16.4 se ejecutará en ambos serví~dores.

Ustado 18.4. PHP puede desencadenar autentic:acl6n HTTP básica.

```
c?php

// Si el u•uario ••ti utilizando IIS, nece•itUIO• ••tatÍlecer $PHP_AUTR_USBR
// y $PHP_AIJTII_PW
lf (•ub•tr($ SERVBR('SBRVBR SOFTWARE'),°· 91 •• 'Nicro•oft• ••
   !l••etl$ SBRV.R('PRP AUTR USBR'II ••
   !isaet($-SBRVER('PRP-AIITII- PII')) H
   •ub•tr($=SBRVER('RTTP_AIJ'l'HORIZATI08'), 0, 61 •• 'B••ic '
    1

   list($ SBRVER['PRP AUTR USBR'), $ SBRVBR[',Sp AUTR Pll')I •
     explode(•:', ba•e'4_decode(•ub•tr($_SBRVBR(-;RTTP:AUTRORIZATIOll'.I,6) 111·


// Sustituya ••to •l •• trata de una in•trucci6n con UD& ba•e de dato•
// o dailar
if ($_SBRVBR('PBP _AUTII_usa•) 1• 'u••r' ll $_SIRVBR['PRP_AUTR_PII') I• .¡¡,.••. ,
(
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
   // el vieitante no ha introducido aue dato• o eu
   // DOllbre y contraaella noeon correcto•

   beader('WWW·Authanticate, Baeic real..•Real••llalle"'ll
   if (aulletr(S SIJlVSll['SSRVIIR sornrARS'I, o, 91 •• 'Nicroeoft'I
      beader(•Statue, 401 Unautborised.fl
   elee
      beader(•HTTP/1,0 401 U'llautborised'I;

   ecbo 'cbl>Go Away!c/bl>';
   ecbo 'You are not autborised to vi- thie reeource. '
)
eln
(
   // el viaitante ba introducido loe dato• correcto•
   ecbo 'cbl>Here it ielc/bl>'I
   ecbo •cp>I llet you are glad you cen eee tbie eecret page.c/p>';

7>
```

El código del listado 16.4actúa de forma muy similar a la de los listados anteriores del capítulo. Si un usuario no ha introducidola información de actualización, se le pedirá. Si los datos introducidos son incorrectos, se devolverá un mensaje rechazándolos. Si incluye un nombre de usuario y una contrasei\a correctos, se devolverú\ los contenidos de la p4gina. El usuario verá una interfaz algo diferente a la devuelta por los listados anteriores. No suministramos un formulario HTML para la informaciónde inicio de sesión. El navegador del usuario mostrará un cuadro de diálogo. Para algunas personas, será una mejora mientras que otros preferirán disponer de control completo sobre los aspectos visuales de la interfaz. En la figura 16.4 se presenta el cuadro de diálogo de inicio de sesión que devuelve Internet Explorer.

Figura 11.4. El navegador del usuario ae enca,ga ele la epartencia del cuadro de «Mlogo al utlllzar la autenUcac16n HTTP.

Como el proceso de autenticación se delega en las funciones int~radas en el navegador, &te opta por ejercer cierta discreción al procesar los intentos de autorización fallidos. Internet Explore, permite al usuario realizar la autenHcación tres veces antes de mostrar la f>'gina de rechazo. Netscape ~tid et~! .el usuario intente autenticane las veces que lo desee (eittie los distintos inteJttos, mostran una cuadro de di61ogo indicando que la autorización ha fallado y preguntará al usuario si desea intentarlo de nuevo). Netsca~ ~lo muestra ~ f>'gina de rechazo si el usuario hace clic en Cmlcelu. Como en el caso del código de los listados 16.1 y 16.2,podemos incluir el código del listado 16.4 en las f>'ginas que d~m,"5 proteger o adjuntarlos al principio de todos los an:hivos de un directorio.

Utilizar autenticación básica con los archivos .htaccess de Apache

Podemos obtener resultados muy similares a los de la secuencia de c,..,mandos anterior sin escribir una secuencia de comandos de PHP.

El servidor Web Apache incorpora una gran cantidad de módulos de autenticación diferentes que se pueden utilizar para determinar la validez de los d.1t,ls introducidos por un servidor. El más sencillo de utilizar es mod_auth, que compara pares de nombre y contrasei\a con las lfneas de un an:hivo de texto incluido en el servidor. Para lograr el mismo resultado de la secuencia de comand,'lS .,nterior, necesitamos crear dos an:hivos HMLT distintos, uno para el contenido y ,ltro para la página de recham. En los ejemplos anteriores nos saltamos algunos elem~ntos de HTML. entre ellos las etiquetas chtal> y cbody>, que deberíamos incluir .ll generar HTML El listado 16.5 recoge el contenido que ven los usuarios autorizados. Hemos llamado a este an:hivo content .html. El listado 16.6 recoge l.l p.igina de error, que hemos llamado rejection.htal. La visualización de esta p.igina es -opcional, pero incorpora un toque profesional si se incluye información útil en ella. Como la página se mostrad cuando un usuario intente entrar en un aireapr,..,tegida, el contenido dtil podrfa incluir instrucciones sobre cómo registrarse p,tra ,lbtener una contrasda o cómo recuperarla y que nos la envíen por coneo electrónico si la hemos olvidado.

Us.. clo 11.5. c:ontent.html. Contenido de ejemplo.

```
cbtal>cbody>
chl>Here it ielc/bl>
cp>I bet you ere glad you can aee tbie eecret page.c/p>
c/body>c/html>
```

Listado 11.1. rejectlon.html. P6gina de enor de ejemplo.

```
cbtal>cbody>
chl>Go Awaytc/hl>
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
cp>You are DOt auellorised to view thia raaourca.c/p>
c/body>c/bt•l>
```

Estos archivos no contienennada nuevo. El código verdaderamente interesante para este ejemplo se recoge en el listado 16.7. Este archivo se puede denominar .htaccess y controlaráel acceso a los archivos y a todos los subdirectorios incluidos dentro de su directorio.

u.t.do 11.7. htaccess. Un archivo .htaccess puede establecer muchos parémetros de configuración de Apache, incluyendo la activación del proceso de autenticación.

```
BrrorDocUIMIDt 401 /chaptar16/rejactioo.html
AuthUaerPile /homa/book/.htpaaa
AuthGroupPila /dav/Dull
AuthRama •Real•••..•
AuthTypa Baaic
require valid-uaer
```

El listado 16.7 es un archivo .htaccess para activar un proceso de autenticación búico en un directorio. En un archivo .htaccesa se pueden configurar muchos parámetros,pero las seis lineas del utilizado en este ejemplo se relacionancon el proceso de autenticación.

!-8 primera línea

```
Brrorqpc-ot 401 /chaptar16/rej•!=tion.html ---------
```

indica a Apache qué documento mostrar para aquellos visitantes que no logren autenticarse(el error H'ITP 401). Puede utilizar otras directivas ErrorDocument para sumútistrar páginas personalizadaspara otros errores HTIP como el 404. Su sintaxis es la siguiente:

```
ErrorDocUIMIDt na.ero_da_error URL
```

Para que una página procese el error 401, es importante que el URL dado esté disponible públicamente.No resulta muy útil sumútistrar una página de error personalizada para indicar a los usuarios que ha fallado su autorización si la página está bloqueada en un directorio en el que necesitan autenticarsepara poder verlo.

La línea

```
AuthUaarPila /bome/book/.htpaaa
```

indica a Apache dónde encontrar el archivo que contiene las contraseñas autorizadas de los usuarios. Este archivo se suele denominar •htpass pero puede asignarle el nombre que desee. Lo importante no es el nombre sino el lugar utilizado para almacenarlo. No debería almacenarse dentro del árbol Web ya que se podría descargara través del servidor Web. En el listado 16.8se recoge el archivo •htpaaa de ejemplo.

Ademú de especificar los usuarios individuales que están autorizados, es posible especificar que sólo aquellos autorizados que se incluyan en determinados grupos dispongan de acceso a los recursos. En nuestro caso no lo hemos hecho, por lo que la línea


```
AuthGroupPila /dav/null
```

establece que el archivo AuthGroupFile apunte a /dev/null, un archivo especial de los sistemas Unix cuyo valor queda garantizado como nulo.

Como el ejemplo PHP, para utilizar autenticación HITP, necesitamos aplicar un nombre a nuestro dominio de la siguiente forma:

```
AuthRa.. "HOllbre-Dominio•
```

Puede seleccionar el nombre de dominio que desee, pero recuerde que este nombre se mostrará a loa visitantes.

Como se admiten varios métodos de autenticación, necesitamos especificar el que estamos utilizando.

Vamos a utilizar la autenticación de tipo Baa ic como se especifica en la siguiente directiva:

```
AuthType Baaic
```

Necesitamos especificar quién dispone de acceso. Podemos especificar usuarios o grupos concretos, o como hemos hecho, todos los accesos de usuarios autenticados. La linea

```
require valid-user
```

especifica que se conceda acceso a cualquier usuario válido.

LlsDdo 11.8. htpass. El archivo de contraaetlas almacena nombres de usuario y la contrasetla cifrada de cada usuario.

```
uaar1:0DRp9M80GS7zM
uaar2:nCl3a<>TOhp.ow
uaar3:yjQMCPlfjXPTzU
uaar4:LOllMEi/bAme2
```

Cada linea del archivo .htpaas contiene un nombre de usuario, dos puntos y la contrasefta cifrada del usuario.

Los contenidos exactos del archivo •htpaaa variarán. Para crearlo, se utiliza un pequeño programa llamado htpaa-d que se incluye en la distribución de Apache.

El programa htpasswd se utiliza de una de las siguientes formas:

```
htpaaawd l-c:mdpaJ arcbivo_de_contr•••.ftaanombra_da_uauario
```

o

```
htpaHwd ·b[c:mdpal arcJJivo_da_cantraadaa.DGllbra_da_uauario contraaa4•
```

El único modificador que necesita utilizar es -c. Este modificador indica a htpaaawd que cree el archivo. Debe utilizarlo para el primer usuario que agregue.

EL_HOLISTICO@HOTMAIL.COM - 2006

Debe tener cuidado de no utilizarlo para el resto de los usuarios porque si el arehívo existiese, htpasswd lo eliminarla y crearía uno nuevo.

Los modificadores opcionales m, d, p o• se pueden utilizar si desea especificar el algoritmo de cifrado que desea utilizar (incluida la opción de no utilizar cifrado alguno). El modificador b indica al programa que espere la contrasei\a como parámetro, en lugar de solicitarla, lo cual resulta útil si desea llamar a btpasswd de forma no interactiva como parte de un proceso por lotes, pero no debería utilizarse si está llamando a btpaHwd desde la lfnea de comandos.

Los siguientes comandos crean el archivo que se muestra en el listado 16.8:

```
htpaaawd -be /home/book/,htpaaa uaarl paaal
htpaaawd ·b /home/book/.htpaaa uaar2 paaa2
htpaaawd -b /il0lMl/book/.htpaaa uaar4 paaal
htpaaawd -b /bolMl/book/.htpaaa uaar4 pa•••
```

Este tipo de autenticación resulta sendllo de configurar, pero el uso de un archivo . htacceH de esta forma plantea una serie de problemas. Los usuarios y las contrasellas se almacenan en un archivo de texto. Cada vez que un navegador solicita un archivo protegido por el archivo . htacceas el servidor debe analizar di('ho archivo y, a continuación, el archivo de contrasei\as, para determinar la coincidencia del nombre de usuario y la contraseña. En lugar de utilizar un archivo •htaccess, podríamos especificar las mismas acciones en el archivo httpd. conf, el archivo de configuración principal para el servidor Web. Cada vez que se solicita un archivo se analiza un archivo . btaccess. El archivo bt tpd. conf sólo se analiza cuando el servidor se abre inicialmente. Este método resulta más rápido pero obiiga a detener y reiniciar el servidor si necesita realizar cambios.

Independientemente de dónde almacene las directivas de servidor, será necesario comprobar la contrasella en el archivo de contraseñas de cada petición. Esta técnica, como en el caso de las vistas anteriormente en las que se utiliza un archivo sin procesar, no resulta adecuada cuando se manejan cientos o miles de usuarios.

Autenticación básica con 115 ...

Como Apache, US admite autenticación HTIP. Apache utiliza el enfoque de UNIX y se controla mediante la edición de archivos de texto. En el caso de 115, la configuración se realiza a través de cuadros de diálogo.

Windows XP permite cambiar la configuración de Internet Information Server S (1155) utilizando el Administrador de HMClos de Internet Esta utilidad se incluye dentro de la sección Herramientas administrativas del Panel de control.

El Administrador de servicios de lntemet presenta un aspecto parecido al que se recoge en la figura 16.5. El control de árbol situado en la parte izquierda muestra que el Krvidor windowa- aerver incluye varios servicios. En el que estamos interesados es el sitio Web predeterminado. Dentro de estos sitio Web, tenemos un directorio llamado protected. Dentro de este directorio hay un archivo llamado content. btml.

Figura 11.S. La consola de admlnislracl6n de MlcnJeaft permlle conligl.nr lnternet lnformatlon Servltr 5.

Para agregar autenticación básica al directorio protected, haga clic con el botón derecho del ratón sobre él y seleccione Propiedades en el menú contextual. El cuadro de di6logo Propiedades permite cambiar una gran cantida6*¡.arámetros del directorio. Las dos fichas en las que estamos interesados son ~urlded de directorios y Errores personalizados. Una de las opciones de la ficha de Seguridad de directorios es Control de autenticación y acc:eso anónimo. Si hace clic sobre el botón Modiflcu se abrirá el cuadro de diálogo que se muestra en la figura 16.6.

Figura 11.1. IIS5 permite et ac:ceso anónimo de manera pradelermlnada, pero nos permile acllvar la eulenllcación.

EL_HOLISTICO@HOTMAIL.COM - 2006

Dentro de este cuadro de dWogo, podemos deshabilitar el acceso anónimo y activar la autenticadón básica. Con los pmmetros que se muestran en la figura 16.6, sólo aquellos usuarios que suministren un nombre y una contraseña adecuados pueden ver los archivos de este directorio.

Pará poder duplicar el comportamiento de los ejemplos anteriores, incluiremos una página para il'ldicar a los usuarios que sus detalles de autenticación no son correctos. Al cerrar el cuadro de diálogo de los mftodos de autenticación podremos seleccionar la ficha Errores personalizados.

La ficha Errores personalizados, ilustrada en la figura 16.7, asoda errores a mensajes de error. En este caso, hemos almacenado el misme archivo de rechazo utilizado anteriormente, rejection.html, ilustrado el listado 16.6. IIS permite incluir mensajes de error más específicos que los que permite Apache, con el código de error HTl'P que tuvo lugar y una explicac:i6n. Podemos incluir diferentes mensajes para cada caso, pero en nuestro ejemplo hemos optado por sustituir los dos que van a tener lugar en este ejemplo con nuestra página de error.

Figura 11.7. La ficha Enores penanalludos permite uocis páginas de error a evenlos de error.

Esto es todo lo que necesitamos para implementarla autenticación de este directorio utilizando DSS. Como ocurre con gran parte del software de Windows, resulta más sencillo de configurar que el software paralelo de Unix, pero más diffdl de copüir de un equipo a otro o de un directorio a otro. Tambim resulta más sendllo equivocane al configurarlo y hacer que el equipo resulte inseguro.

El peor defecto de IIS es que para autenticar a los usuarios Web compara los datos de registro con cuentas incluidas en el equipo. Si queremos permitir que el

usuario •j ohn • se registre con la contrasella •pasaword•, necesitamos crear una cuenta ?e usuario en el equipo o en un dominio con su nombre y contrasei\a. Es necesano poner mucha atención al crear cuentas para procesos de autenticadón Web de forma que los usuarios dispongan únicamente de los derechos de cuenta que necesiten para ver páginas Web y ningdn otro derecho como acceso Telnet.

Utilizar autenticación mod auth mysql

Como ya se ha indicado, el uso demod_authconApachees unmitodosencillo de configurar y eficaz. Sin embargo, como almacena los sitios en un archivo de texto, no resulta práctico para sitios con tráfico que reciban la visita de una gran cantidad de usuarios.

Afortunadamente, puede disfrutar de la misma fac:ilidad de uso que mod auth y de la velocidad de una base de datos utilizando mod auth myaql. Este módulo func:ionaprácticamente igual que mod_auth, pero como utiliza una base de datos MySQL en lugar de un archivo de texto, permite buscar sobre una larga lista de usuarios rápidamente.

Para utilizar este módulo, es necesario compilarlo e instalarlo en su sistema o pedirle a su administrador de sistemas que lo instale.

lnstalar mod_auth_mysql

Para poder utilizar mod_auth_mya ql, necesitará configurar Apache y MySQL si~~o las ~truccio~es inc:luidas en el Apmdic:e A con uña serie de pasos adiaonales. Las mstrucaones de los archivos READMB y OSAGB de esta distribución resultan bastante buenas. A continuación, inc:luimos un resumen de las mismas.

1. Obtenga el archivo de distribución correspondiente al módulo. Se encuentra inc:luido en el CD-ROM que se adjunta al libro, pero puede obtener la última versión de http://sourceforge.net/project•/mod·auth·mysql/.
2. Descomprima el código fuente.
3. Cambie al directorio mod_auth_myaql, ejecute make y, tras ello, 111ake install. Puede que tenga que modificar la ubic:ac:ión de instalación de MySQL en este archivo.
4. Añada la siguiente linea a httpd. eonf:

```
Loadllodule my9ql_auth_ll0dule libezec/ll0d_auth_my9ql.80
```

para cargar dinámicamente el módulo en Apache.

S. Cree una base d, datos y una tabla en MySQL para inc:luir la información de autenticadón. No tiene por qué tratarse de una base de datos o tabla düeren-

EL_HOLISTICO@HOTMAIL.COM - 2006

te; puede utilizar una tabla existente como la base de datos auth del ejemplo utilizado anteriormente.

6.--Agregue una línea al archivo ht tpci. conf para suministrar al módulo mod_auth_myaql los parámetros que necesita para establecer la conexión a MySQL. La directiva presentará este aspecto:

```
Auth_MySOL_Info boat..... u•er pa••word
```

lfuncionó?

La forma más sencilla de comprobar si la compilación funciona es ver si se.inicia el servidor Apache. Para iniciar Apache, escriba lo siguiente en caso de incorporar compatibilidad SSL:

```
/u•r/local/apacbe/bin/apacbectl •tart••l
```

Si no incorpora compatibilidad SSL, escriba

```
/uer/local/apacbe/bin/apachectl •tart
```

Si se inicia con la directiva Auth_MySQL_Info en el archivo httpd. conf, el módulo mod_auth_myaql se agregará satisfactoriamente.

Utiiiz.ir mod auth mvsql

Tras instalar el módulo, su uso no resultará más complicado que el demod_auth. El listado 16.9 muestra un ejemplo del archivo .htacceBS que autenticará usuarios con contraseftas cifradas almacenadas en la base de datos creada anteriormente en este capítulo.

Llagdo 11.1. El archivo .htacc:esaautenUca usuaitoa utilizando una bate de datos MySQL.

```
lrrorDocument t0l /cbapter1,tr•jection.ht•l

Authllue •Real•••-·
AutbType Baeic

Auth_NySQL_DB autb
Autb_MySQL_Bncryption_Typee MySQL
Auth_MySOL_Paeeword_Table auth
Autb M)'SOL Ueername Pield name
Auth~:,iysoL:Paeeword:rield pa••

require valid·ueer
```

Como puede observar la mayor parte del listado 16.9 es igual a la del listado 16.7. Queda por especificar ün documento de error para cuando falla la autenticiín (error 401). Tambiffl especificamosautenticación básica y el nombre del dominio. Como ocurre con el listadb 16.7, permitimos el acceso de cualquier usuario válido y autenticado.

Como estamos utilizando el módulo mod_auth_myaqly no queremos utilizar todos los parámeb'OSpredeterminados, disponemos de algunas directivas para especificar cómo deberla funcionar. Auth_MySQL_DB, Auth_MySQL_Paaaword_Table,Auth_MySQL_Uaernaiae_Fieldy Auth_MySQL_Paaaword_Field especifican el nombre de la base de datos, la tabla, el campo de nombre de usuario y el campo de contraseila, respectivamente.

Hemos incluido la directiva Auth_MySQL_Bncryption_Typea para especificar que queremos utilizar el cifrado de contrueilas MySQL. Los valores aceptables son Plaintext, Crypt_DBS o NySQL. Crypt_DBS (el predeterminado} utiliza el sistema de contrasei\as ci&adas DES estmdar de UNIX.

Desde la perspectiva del usuario, este ejemplo del módulo iaod_auth_myaql funciona exactamente igual que el e¡emplo mod_auth. El navegador Web le presentará un cuadro de diálogo. Si la autenticación resulta satisfactoria, se mostrará el contenido. Si la autenticación falla, se devolverá la página de error.

Para muchos sitios Web,elm6duloiaod_auth_mysql es ideal. Esnpido, relativamente sencillo de implementar y permite utilizar cualquier IIMlllúsmo que resulte práctico para agregar entradas de base de datos para nuevorasuarios. Para obtener más flexibilidad y la posibilidad de lograr un control más eJChaustivosobre secciones de las páginas, puede implementar su propio sistema de autenticación utilizando PHP y MySQL

Crear un sistema de autenticación ...ri"iio

Hemos visto cómo crear nuesb'OSpropios métodos de autenticación incluyendo algunos defectos y ciertos compromisos, y hemos utilizado métodos de autenticación incorporados, que resultan menos flexibles que escribir nuestro propio código. En un capítulo posterior, cuando abordemos el tema del control de sesiones, podremos escribir nuestro propio sistema de autenticación con menos compromisos que en este capítulo. En un capítulo posterior, desarrollaremos un sistema de autenticación simple de usuario que evite algunos de los problemas vistos aquí utilizando sesiones para realizar el seguimiento de variables entre páginas.

En otro capítulo posterior, aplicaremos este enfoque a un proyecto del mundo real y veremos cómo se puede utilizar para implementar un sistema de autenticación más exhaustivo.

$name();

Los detalles de la autenticación HTTP se especifican en la RCF 2617, que está disponibleenhttp://www.rfc-editor.org/rfc/rfc2617. txt.

EL_HOLISTICO@HOTMAIL.COM - 2006

La doc:umenmción de mod_autll. que controla **la** autenticación búica en Apache, puedeencontnrlaenhttp://www.apache.org/docs/mod/mod_auth.html .
La doaunentación del módulo mod_auth_mysql se incluye en el archivo de descarga. El tamu'\o de este archivo es pequeilo, por lo que merece la pena descargarlo y conaultar el archivo readme.

A continuación

En el siguiente capítulo se explica cómo salvaguardar las datos en todas las fases de procesamiento, desde su entrada, has~ su transmisión y almacenamiento. Incluye el uso de SSL, certificados digitales y cürado.

EL_HOLISTICO@HOTMAIL.COM - 2006

En este capítulo, vamos a explicar cómo tratar de forma segura los datos de usuario desde la introducción y transmisión hasta la fase de almacenamien~, lo que nos permitirá implementar una transacción entre nosotros y un usuario de forma segura de extremo a extremo. En este capítulo nos centraremos en los siguientes aspectos:

- Suministrar transacciones seguras
- Utilizar SSL (Secure Sockets Layer, Capa de sockets segura)
- Almacenamiento seguro
- Determinar si almacenar o no números de tarjeta de crédito
- Utilizar encriptación en PHP

Transacciones seguras

Para garantizar el suministro de transacciones seguras a través de Internet es necesario examinar el flujo de información de nuestro sistema y garantizar la seguridad de la información en cada punto. En el tema de la seguridad de Internet, no hay conceptos absolutos. Ningún sistema tiene garantizada su impenetrabilidad para siempre. El término seguro indica que el nivel de esfuerzo aplicado a un sistema o transmisión es alto comparado con el valor de la información.

EL_HOLISTICO@HOTMAIL.COM - 2006

Si queremos dirigir nuestros esfuerzos de seguridad de manera eficaz, necesitamos examinar el flujo de la información a través de todos las partes del sistema. En la figura 17.1 se ilustra el flujo que sigue la información de usuario en una aplicación típica escrita utilizando PHP y MySQL

Figura 17.1. La información de usuario es almacenada o procesada por los siguientes elementos de un entomo de aplicación Web tlplco.

Los detalles de cada transacción que tenga lugar en su sistema pueden variar en función del diseño, de los datos del usuario y de las acciones que desencadenaron la transacción. Puede examinar todos estos elementos de una manera similar. Cada -J,ansacción en~~.!21!caci°!'.W.~b_y_~n~uario C.0!l'iffl.za con el !_!!Vfo de una solicitud desde el navegador del usuario a través de Internet hasta el servidor Web. Si la página es una secuencia de comandos de PHP, el servidor Web delegará el procesamiento de la página en el motor PHP.

La secuencia de comandos de PHP puede leer y escribir datos en el disco. También puede incluir o requerir otros archivos de PHP o HTML Así mismo puede enviar consultas de SQL al demonio de MySQL y recibir respuestas. El motor de MySQL es responsable de leer y escribir sus propios datos en el disco.

Este sistema tiene tres partes fundamentales:

- El equipo del usuario
- Internet
- El sistema

A continuación, analizaremos los aspectos de seguridad relacionados con cada una de estas partes de manera separada, pero obviamente el equipo del usuario e Internet son las que quedan más lejos de nuestro control.

El equipo del usuario-

Desde riuestro punto de vista, el equipo del usuario ejecuta un navegador Web. No disponemos de control sobre factores como la seguridad aplicada al equipo.

Debemos tener en cuenta que el equipo puede resultar muy poco seguro o que se trate tle un terminal compartido en una biblioteca; escuela o café.

Existen muchos navegadores diferentes en el mercado y cada uno de ellos incorpora funciones diferentes. Si consideremos únicamente las versiones más recientes de los dos navegaderes más populares, las innumerables diferencias entre ellos afectarán a la forma de generar y representar el código HI'ML. a lo que hay que ailadir las cuestiones de seguridad y funcionalidad.

Debemos tener en cuenta que muchos usuarios deshabili~rán funciones que consideren como un riesgo para su seguridad o privacidad, como las funciones de Java, las CC>Qkies o JavaScript. Si utilizamos estas funciones, deberíamos comprobar que nuestra aplicación no pierde demasiado atractivo para los usuarios sin ellas o considerar el uso de una interfaz menos completa que les permita utilizar nuestro sitio.

Los usuarios situados fuera de los EEUU y Canadá puede que utilicen navegadores que sólo admitan un sistema de encriptación de 40 bits. Aunque el gobierno de los EEUU ha cambiado la ley en enero de 2002 para permitir la exportación de encriptación segúra (excluyendo a aquellos países sobre los que pese un embargo) y las versione, de 128 bits ya están disponibles para la mayoría de los usuarios, es probable que muchos no hayan actualizado sus sistemás. Si no incluye garantías de seguridad a los usuarios de manera textual en su sitio, este hecho no debería ser motivo de preocupación para los desarrolladores Web ya que SSL se encargará de procesar automáticamente el nivel de seguridad más alto que pueda establecer enb:e el senddor y el aavegadot' del ns11ario -· -·-·---.

No podemos estar seguros de que la conexión a nuestro sitio Web se produzca desde un navegador Web utilizando la interfaz desarrollada para tal efecto. Las peticiones que llegan a nuestro ~itio pueden proceder de otro sitio qut! esté utilizando imágenes o contenidos sin permiso de su propietario, o de una persona que esté utilizando software como cURL para evitar las medidas de seguridad.

En un capítulo posterior examinaremos la biblioteca cURL, que se puede utilizar para simular conexiones desde un navegador. Como programadores, esta biblioteca nos resulta de utilidad pero recuerde que se puede utilizar de manera malintencionada. Aunque no podemos cambiar o controlar la forma en la que están configurados los equipos de nuestros usuarios, debemos tenerlos en cuenta. La variedad de equipos utilizados por los usuarios para conectarse a Internet puede convertirse en un factor importante a la hora de decidir cuánta funcionalidad suministrar a través de secuencias de comandos del lado del servidor (como PHP) y cuánta a través de secuencias de comandos del lado del diente (como JavaScript).

La funcionalidad suministrada por PHP puede ser compatible con el navegador de todos los usuarios ya que el resultado final no es más que una página HMTL El uso de cualquier otro elemento que no sea código JavaScript básico exigirá tene~en cuenta las diferentes funciones de cada versioo. de los navegadores.

Desde la perspectiva de la seguridad, conviene utilizar secuencias de comandos del lado del servidor para operaciones como la validación de datos porque, de esta forma, nuestro código fuente no resultará visible para el usuario. Si sólo validamos los datos en JavaScript, los usuarios podrán ver el código y qu~ evitarlo.

EL_HOLISTICO@HOTMAIL.COM - 2006

Los datos que resulte necesario guardar podemos almacenarlos en nuestros propios equipos, como archivos o registros de base de datos, o en los equipos de nuestros usuarios en forma de coolcies. En un capítulo posterior veremos cómo utilizar coolcies para almacenar datos restringidos (como una clave de sesión).

La mayor parte de los datos que almacl!hamos deberían alojarse en el servidor Web o en nuestra base de datos. Existen buenas razones para almacenar la menor cantidad posible de datos en el equipo del usuario. Si la información no se almacena en nuestro equipo, no dispondremos de control sobre la seguridad de su almacenamiento, no podremos estar seguros de que el usuario no la eliminará ni podremos impedir que la modifique para intentar confundir al sistema.

Internet

Al igual que ocurre con el equipo del usuario, tenemos muy poco control sobre las caracteristicu de Internet, pero eso no significa que tengamos que ignorarlu al diseftarelsistema.

Internet incorpora una gran cantidad de funciones interesantes, pero es un lugar inseguro de mane~ inherente. Al enviar información desde un punto a otro, es necesario tener en cuenta que otros podrían verla o alterarla, como se comeató en un capítulo anterior. Con esto en mente, puede decidir quf acción tomar.

Podría optar por:

- Transmitir la información de todas formas, sabiendo que podría leerse.
- Cifrar o firmar la información antes de transmitirla para mantener su privacidad o protegerla ante posibles manipulaciones.
- Decidir que la información es demasiado valiosa para arriesgarse a divulgarla y buscar otra forma de distribuirla.

Internet es tambifn un medio butante anónimo. Resulta dificil determinar si una persona con la que mantiene relación resulta ser quifn dice ser. Incluso en el caso de que podamos aseguramos de que un usuaria es quifn dice ser, resultaría diffcil demostrarlo sin ninguna duda en un foro público como un tribunal, lo que puede dar lugar a problemu como el del repudio, visto en un capitulo anterior.

En definitiva, las transacciones en Internet presentan dos problemu fundamentales, la privacidad y el repudio.

Existen al menos dos formas diferentes de garantizar el flujo de la información desde nuestro servidor Web hasta Internet y vicevena:

- SSL (Capa segura de sockets)
- S-HTIP (Protocolo de transferencia de hipertexto seguro)

Ambas tecnologfas ofrecen mensajes privados y no manipulables, y autenticación, pero SSL está más extendido y resulta más sencillo de implementarmientru que S-HTIP no acaba de calar. En este capítulo examinaremos SSL.

Su sistema

La parte del universo sobre la que disponemos de control es nuestro sistema. Nuestro sistema viene representado por los componentes que se incluyen dentro del cuadro ilüstJ:ado en la figura 17.1. Estos componentes pueden estar separados ffsicamente en una red o incluirse en un único equipo físico.

Resulta butante seguro despreocuparse de la seguridad de la información si delegamos esta tarea en productos de terceros. Es probable que los autores de estu aplicaciones hayan dedicado más tiempo al tema que nosotros. Mientru utilice una veni6n actualizada de un producto reconocido, podri descubrir cualquier problema conocido mediante la acertada aplicación de Google o su motor de búsqueda preferido. Deberla uignar múima prioridad a la tarea de mantener esta información al día.

Si es responsable de los procesos de instalación y configuración, deberi preocuparse por la forma de instalar y configurar el software. Muchos errores de seguridad surgen como resultado de no seguir las advertencias de la documentación o implican aspectos relativos a la administración general del sistema. Cómprese un buen libro sobre la administración del sistema operativo que tenga la intención de utilizar o contrate los servicios de un administrador de sistemu con experiencim.

Al instalar PHP deberla tener en cuenta que resulta más seguro y más eficiente hacerlo en forma de un módulo SAPI para el servidor Web que ejecutarlo a travé de la interfaz CGI.

Un aspecto que debe considerar es quf pueden hacer sus secuencias de comudos. Pregúntese quf datos potencialmente delicados trasmite su aplicación al usuario a travfs de Internet y qui datos estamos pidiendo a nuestros usuarios que nos envíen. Si estamos transmitiendo información entre nosotros y nuestros usuarios cuya privacidad debamos garantizar o cuya manipulaáón debamos dificultar, deberíamos considerarla opción de utilizar SSL.

Ya hemos hablado del uso de SSL entre el equipo del usuario y el servidor. Tambifn deberíamos prever la situación en la que se transmitan datos desde un componente de nuestro sistema a otro a travfs de la red. Un ejemplo tfpico seria si nuestra base de datos estuviera alojada en un equipo diferente a nuestro servidor Web. PHP se conectarla al servidor MySQL a travfs de TCP/IP y esta conexión no estaría encriptada. Si los equipos estuvieran en una red de área local privada, debería asegurarse de que la red es segura. Si los equipos se conectan a travfs de Internet, el sistema probablemente resulte lento y necesite tratar la conexión de la misma forma que otras conexiones sobre Internet. Es importante asegurarse de que cuando los usuarios crean estar interactuando con nosotros as(sea. La obtención de un certificado digital protegerá a nuestros visitantes contra prácticu de penoniftcación y nos permitirá utilizar SSL sin que los usuarios vean aparecer un mensaje de aviso además de _aportar un aire de respetabilidad a nuestra aventura en linea.

¿Comprueban minuciosamente nuestru secuencias de comandos los datos introducidos por los usuarios? ¿Tenemos cuidado de almacenar la información de manera segura? En las siguientes secciones responderemos a estu preguntu.

EL_HOLISTICO@HOTMAIL.COM - 2006

Utilizar SSL

El protocolo SSL fue disellado originalmente por Netscape para facilitar la seguridad de las comunicaciones entre servidores y navegadores Web. Desde entonces ha sido adoptado como el método estúldar de manera no oficial por los navegadores y servidores para intercambiar información delicada.

Existe una amplia compatibilidad para las versiones 2 y 3 de SSL La mayor parte de los servidores Web incluyen funcionalidad SSL o pueden aceptarla como módulo complementario. Internet Explorer y Netscape incorporan compatibilidad SSL desde la versión 3.

Los protocolos de red y el software que los implemente se suelen organizar en forma de pila de capas. Cada capa puede pasar datos a la capa superior e inferior, y solicitar servicios de la capa situada por encima o por debajo. La figura 17.2 muestra una pila de protocolos.

Flgura 17.2. La pila de prolOCOloa utilizada por un protocolo de c.pa de apllcacl6n ~ el Protocolo de transferencia de hipeftexto.

C~a;do se utiliza HTI'P para transferir información, el protocolo HTIP llama al Protocolo de control de transmisión (TCP) que, a su vez, se bua en el Protocolo de Internet. Este protocolo necesita un protocolo adecuado para que el hardware de red pueda utilizarse para tomar paquetes de datos y enviarlos como sel\al eléctrica a nuestro destino.

HTI'P es un protocolo de capa de aplicación. Existen muchos otros protocolos de capa de aplicación, como rn, SMTP y Telnet (como se muestra en la figura) y otros como POP e IMAP. TCP es uno de los dos protocolos de capa de transporte utilizados en redes TCP/IP. IP es el protocolo de la capa de red. El host hasta la capa de red es responsable de conectar nuestro host (equipo) a una red. La pila de prc,tocolos TCP/IP no especifica los protocolos utilizados para esta capa, ya que necesitamos diferentes protocolos para los diferentes tipos de redes.

Al enviar datos, éstos se dirigen a través de la pila desde una aplicación a un dispositivo ffsico de red. Al recibir los datos, éstos se dirigen desde la red ffsica, a través de la pila, hasta la aplicación.

El uso de SSL agrega una capa transparente adicional a este modelo. La capa SSL eXIste entre la capa de transporte y la capa de aplicación, como se ilustra en la figura 17.3:La cap~ SSL modifica los datos desde nuestra aplicación HTI'P antes de pasarlos a.la capa de transporte para enviarlos a su destino.

SSL es teóricamente capaz de crear un entorno de transmisión seguro para otros protocolos aparte de HITP, pero se suele utilizar para HTI'P. Se pueden utilizar otros protocolos porque.la capa SSL es búicamente transparente. Esta capa propor-

dona la misma interfaz a los protocolos situados por encima que la capa de transporte subyacente. A continuación se encarga, de forma transparente, de los saludos, de la encriptación y de la decriptación.

Figura 17.3. SSL egrega una capa adicional a la pila de protoc:oloa asl como protocolos de capa de aplicación para controlar sus propias operaciones.

Cuando un navegador Web se conecta a un servidor Web seguro a través de H'ITP, los dos necesitan seguir un protocolo de saludo para llegar a un acuerdo sobre elementos como la autenticación y la encriptadón.

La secuencia de saludo implica los siguientes pasos:

1. El navegador se conecta a un servidor con SSL habilitado y solicita que se autentique.
2. El servidor envfa su certificado digital.
3. El ~rvid9rp~~e !lC>lici taro_pcio~llllll!J!te.!~l:'"q~~-nosuele ser lo normal) al navegador que se autentique.
4. El navegador presenta una lista con los algoritmos de encriptación y funciones hash que admite. El servidor selecciona la encriptadón más segur!' que admite.
5. El navegador y el servidor generan claves de sesión:
   a. El navegador obtiene la dave pública desde su certificado digital y la utiliza para encriptar un número generado de forma aleatoria.
   b. El servidor responde con m'5 datos aleatorios en forma de texto sin procesar (a menos que el navegador haya suministraclo un certificado digital en la solicitud del servidor en cuyo caso el servidor utiliza la clave •pública del navegador).
   c. Las claves de encriptación para la sesión se generan a partir de estos datos aleatorios utilizando funciones hash.

El proceso de saludo no es instantáneo porque la generación de datos aleatorios de calidad, la decriptadón de los certificados digitales, la generación de las claves y el uso de la criptograffa de clave pública lleYa su tiempo. Afortunadamente, los resultados se guardan en caché, de manera que si el mismo navegador y servidor desean intercmnb~r varios mensajes seguros, el proceso de saludo y el tiempo de procesamienlJOrequerido sólo tendrm lugar una vez.

EL_HOLISTICO@HOTMAIL.COM - 2006

Cuando los datos se envfan a traws de una conexión SSL, tienen lugar _los siguientes pasos:

1. Se dividen en paquetes que resulten '!"cilios de procesar.
2. Cada paquete se comprime (opcionalmente).
3. Cada paquete lleva asignado un código de autenticación de mensaje (MAC) calculado utilizando un algoritmo hash.
4. El código MAC y los datos comprimidos se combinan y se encriptan.
5. Los paquetes encriptados se combinan con información de cabecera y se envfan a la red.

En la figura 17.4 se ilustra todo el proceso.

Flgunl 17.4. SSL divide, comprime, aplica una función hah a los datol y loa enc:rtpta antes de enviarlos.

Como observar, en el diagrama, el encabezado TCP se agrega tras encriptar los datos, lo que significa que la información dirigida podria manipularse y aunque los husmeadores no puedan ver la información que estamos intercambiando sf podrfan ver quién lo esti haciendo.

La razón por la que SSL realiza la compresión antes de la encriptación responde a que, aunque el trifico de red pueda (como ocurre a menudo) comprimirse antes de transmitirse por una red, los datos encriptados no se comprimen bien. Los sistemas de compresión se basan en la identificación de repetic ioneso patrones dentro de los datos. El intento de aplicar un algoritmo de compresión tras convertir los

datos en un conjunto aleatorio de bits mediante un proceso de encriptación no tiene mucho sentido. Seria una pena-si SSL, que fue disdado para incrementar la seguridad de la red, tuviera el efecto secundario de incrementar el bifico de red.

Aunque SSL es relativamente complejo, los usuarios y los desanolladores resultarin ajenos a gran parte lb! lo que ocurre, ya ~e sus interfaces externas simulan protocolos existentes.

En el futuro, SSL 3.0 podr(a ser sustituido por TLS 1.0 (Seguridad de capa de transporte) pero en el moaierito de escribir éstai lfneas, TLS se encuentra todavfa en la fase de estmdar de borndor y no resulta compatible con la mayoría de los servidores y navegadores. La intención de TLS es la de convertirse en IDI verdadero estindar abierto, en lugar de un estándar definido por una organización que se ha puesto a disposición de otras. TLS se basa directamente en SSL 3.0 pero contiene mejoras pensadu para superar los puntos débiles de SSL.

Filtrar las entradas de los usuarios

Uno de los principios del diseflo de una aplicación Web segura -.ue no se debe confiar nunca en la información introducida por el usuario. Filtre sijinpre los datos introducidos por el usuario antes de colocarlos en un atthivo o battde datos, o de pasarlos a travá de un comando de ejecución del sistema.

A lo largo de este libro hemos hablado varias.veces de las tbicas que se pueden utilizar para filtrar las entradas de los usuarios. A continuac:iónt·los recogemos brevemente para recordarlas.

- La función addalashea () deberla utilizarse para filtrar los datos del usuario antes de pasarlos a una base de datos. Esta función evitan todos los caracteres especiales que podrfan presentar problemas en una base de datos. Puede utilizar la función stripalaahea (I pu;a devolver los datos a su forma original.
- Puedeactivarlasdirectivasmagic_quotea_gpcymagic_quotes_runti• en su archivo php. ini. Estas directivas agregan y-quitan las barras invertidas automiticamente. La directiva magic _quotea _gpc aplica este formato a las variables OBT, POST y cookies entrantes, y la directiva magic_quotes_runti- lo aplicad a los datos dirigidos o procedentes de bases de datos.
- La función eacapeshellc:md () deberfa utilizane al pasar datos del usuario a llamadas ayatem () o exec (), o a apóstrofes invertidos. Esta función evitanl ~os los metacaracteles que se puedan utilizar para obligar al sistema a ejecutar comandos arbitrarios introducidos por un usuario malintencionado.
- Puede utilizar la función strip_tags () para limpiar de etiquetas HTML J PHP una cadena. Esta función evitanl que los usuarios introduzcan secuen-

EL_HOLISTICO@HOTMAIL.COM - 2006

cías de comandos malintencionados en los datos que puedan representarse en el navegador.

- Puede utilizar la función htmlspecialchars(l, que convierte los caracteres en sus equivalentes de entidad HTML Por ejemplo, < se convertirá en <. Esta fundón convierte todas las etiquetas de secuencia de comandos en caracteres inofensivos.

Proporcionar un almacenamiento seguro

Los tres tipos diferentes de datos almacenados (archivos HTML o PHP, datos relacionados con secuencias de comandos y datos MySQL)se almacenarán en áreas diferentes del mismo disco, aunque se muestran de forma separada en la figura 17.1. Cada tipo de almacenamiento requiere adoptar distintas precauciones y se examinará de manera individual.

El tipo de datos más peligroso que almacenaremos es el contenido ejecutable. En un sitio Web, suelen adoptar la forma de secuencias de comandos. Es necesario ponermucho cuidado al establecer los permisos de archivo dentro de la jerarquía Web. En concreto, el árbol de directorios cuyo primer elemento es htdoca en un servidor Apache y inetpub en un servidor IIS. Otros necesitan disponer de permisos 'para leer nuestras secuencias de comandos para poder ver su salida, pero no deberian ~ permilo para IICRbir o ecfüarlos

La'misma advertencia se aplica a los directorios incluidos dentro de la jerarquía Web. Sólo nosotros deberíamos disponer de permiso para escribir en dichos directorios. Otros usuarios, incluido el usuario utilizado por el servidor Web para ejecutarse, no deberían disponer de permiso para escribir o aear nuevos archivos en directorios que se puedan cargar desde el servidor Web. Si permite que otros escriban archivos en este punto, podrían crear una secuencia de comandos malintencionada y ejecutarla al cargarla a través del sen·idor Web.

Si sus secuencias de comandos necesitan disponer de permiso para escribir en los archivos, cree un directorio fuera del árbol Web para ello, en especial si se trata de secuencias de comandos para la carga de archivos en el servidor. No deberían mezclarse las secuencias de comandos y los datos que escriben.

Al escribir datos delicados, es probable que se vea tentado a encriptarlos como primer paso. En este enfoque, sin embargo, no resultará de mucho valor. Veamos por qur. si tememos un archivo llamadonQmerostajetascr6dito. txt en nuestro servidor Web y un pirata informático obtiene acceso a nuestro servidor y logra leerlo, ¿qué podría leer? Para poder encriptar y decriptar los datos necesitamos un programa que los cifre y un programa que los desdhe, y uno o varios archivos de claves. Si el pirata puede leer los datos, nada le impedirá, con toda probabilidad, leer los archivos de clave asf como otros archivos.

La enc!Iiptación de los datos puede resultar valiosa en un servidor Web, pero sólo si el software y las claves para decriptar los datos no se almacenan en el servidor Web sino en otro equipo. Una forma de tratar datos delicados de forma

segura sería encriptarlos en el servidor y transmitirlos a otro equipo, por correo electrónico, por ejemplo.

Los datos de una base de datos son similares a los archivos de datos. Si se configura MySQL de manera correcta, sólo MySQLpodrá escribir en sus archivos de da~, lo cual significa que sólo necesitamos preocupamos por los accesos de los usua~ dentro de MySQL. Ya se ha comentado el sistema de permisos de MySQL, que asigna derechos conaetos a nombres de usuarios concretos en host concretos.

También hay que destacar el hecho de que a menudo necesitaremos escribir una contr~ de MySQLen una secuencia de comandos de PHP. Por regla general, las secuenaas de comandos de PHP se pueden cargar públicamente. Este hecho no es tan peligr~ como pu~era parecer a J?rimeravista ya que si no se penetra en la configuraaon del servidor Web, el código PHP no resultará visible desde el exterior.

Si el servidor Web se configura para analizar archivos con la extensión .php u;'llndo el intérprete de PHP, no se podrá ver desde fuera el código sin interpretar. Sm embargo, conviene tener cuidado al utilizar otras extensiones. Si coloca archivos ·inc en sus directorios Web, cualquier persona que los solicite recibirá el código sin analizar. Por lo tanto, será necesario colocar los archivos de inclusión fuera del árbol Web, configurar el servidor para que entregue archivos en esta extensión o utilizar .php también como extensión en éstos.

Si está compartiendo un servidor Web con otros, las contraseftas de MySQL podrían. resultar visibles a otros usuarios del mismo equipo que puedan ejecutar sec~s d~ comandos~ tra,és del mismo senidor Web. Esta situaci6n puecle-resultar inevitableen funaón de la configuración del sistema. Para evitarla, puede configurar el servídor Web de manera que ejecute secuencias de comandos como usuari~ individuales o que. cada usuario ejecute su propia instancia del servidor Web. 51 no es el administrador del servidor Web(como ocurrirá con toda probabilidad si ~mparte un servidor), es aconsejable que comente esta situación con el administrador y busque opciones de seguridad.

Determinar si almacenar o no números de tarjeta de crédito

Tras comentar el tema del almacenamiento seguro de datos delicados. debemos detenernos en un tipo de datos que merece especial atención. Los usuarios de Internet están especialmente preocupados por el número de sus tarjetas de crédito. Si va a almacenar-estos números, debe prestar especial cuidado. Pregúntese para qué los necesita y si resulta verdaderamente necesario almacenarlos.

¿Qué va hacer con un número de tarjeta? Si el tipo de transaa:iones con el ~uario va a ser esporádico y su sistema realiza el procesamiento dt tarjetas en tiempo real, convendría aceptar el número de tarjeta del diente y enviarlo directa-r,raente a la pasarela de procesamiento de transacciones sin almacenarlo.

EL_HOLISTICO@HOTMAIL.COM - 2006

Si necesita aplicar cargos cada cierto tiempo, por ejemplo aplicar una cantidad mensual a cada tarjeta por una suscripción, esta opción puede que no resulte la mú acertada. En este caso, deberfa considerar la posibilidad de almacenar los números en algún lugar distinto al servidor Web.

Si va a almacenar grandes cantidades de detalles relativos a las tarjetas de clientes, asegúrese de que el administrador del sistema tenga ex~a en el ~ y que disponga de tiempo para comprobar las actualizaciones de la informaaón de seguridad del sistema operativo y de otros productos que utilice.

Utilizar encriptación en PHP

Una acción sencilla pero útil a la que podemos recurrirpara demostrar el uso de la encriptación es enviar un mensaje de correo electrónico cifrado. El estándar de facto utilizado para encriptar correos electrónicos durante muchos ai\os ha sido PGP, que equivale a Pretty Good Privacy (Privacidad bastante buena). Este estútdar fue escrito por Philip R. Zimmermann especialmente para agregar privacidad al correo electrónico.

Existen versiones gratuitas de PGP; pero debe tener en cuenta que no se trata de software gratuito. La versión gratuita sólo se puede utilizar de manera legal con fines no comerciales. Si es ciudadano estadounidense con residencia en los EEUU o ciudadano canadiense con residencia en Canadá, puede conseguir la versión gratuita en http://web.mit.edu/network/pgp.html.

Si desea utilizar PGP de forma comercial, puede obtener una licencia comercial de PGP Corporation. Encontrará mú detalles al respecto en http://www.pgp.com.

Para obtener una licencia de PGP fuera de los EEUU y Canadá, consulte la lista de sitios de descarga internacional en la página de PGP: http://www.pgpi.org.

Recientemente ha surgido una alternativa de código abierto a PGP. GPG (Gnu Privacy Guard) no conti~ algoritmos patentados y se puede utilizar comercialmente sin restricción.

Ambos productos realizan la misma tarea de foftna similar: Si tiene pensado utilizar herramientas de linea de comandos, la elección no importa, pero cada una incorpora interfaces diferentes como complementos para programas de correo electrónico que descifran los mensajes recibidos automáticamente.

GPG est4 disponible en http://www.gnupg.org.

Puede utilizar los dos productos de manera conjunta. Por ejemplo, puede crear un mensaje encriptado con GPG y dirigirlo a alguien que utilice PGP (siempreque sea una versión reciente) para descifrarlo. Como nosotros estamos interesados en la aeación de mensajes en el servidor Web, vamos a ver un ejemplo del uso de GPG. Si se utiliza PGP, la operación no requerirá muchos cambios.

Como en el caso de otros ejemplos de este libro, necesitará disponer de GPG para que el código funcione. Puede que ya tenga instalada esta herramienta en su sistema. De no ser asf, no se inquiete, ya que el procedimiento de instalación resulta sencillo (aunque la confip~ación puede complicarse).

Instalar GP<i

Para agregar GPG a nuestro equipo de Linux, descargamos el archivo pertinente de www.gnupg.org. Necesitaremos utilizar gunaip y tar o extraer los archivos del archivo comprimido, dependiendo de su tipo, •tar. gs o •tar. bs2.

Para compilar e instalar el programa, utilice los mismos comandos que en la mayor parte de las versiones de Linux:

configure (o ./configure en función de su sistema)

make

-ke in•tall

Si no es el usuario rafz, tendrá que configurar la secuencia de comandos con la opción - -prefix de la siguiente forma:

```
./configure --prefix•/ruta/a/eu/directorio
```

ya que los usuarios distintos al usuario rafz no disponen de acceso al directorio predeterminado de GPG.

Si todo discurre satisfactoriamente, GPG se compilará y el ejecutable se copiará en /usr/loc:al/bin/gpg o en el directorio que especifiquemoa.-Puede cambiar muchas opciones. Consulte la documentación de GPG si desea all!tener mú detalles. Para un servidor Windows, el proceso resulta incluso mú sencillo. De8cugue el archivo zip, descomprfmalo y coloqueel archivo gpg. exe en algún punto de su PATH. (C: \Window•\, por ejemplo). Cree un directorio en e: \gnv.pg. Abra el sfmbolo de comandos y escriba gpg.

Tambic!n necesitamos instalar GPG o PGP y generar un par de claves eri el sistema desde las que tengamos previsto comprobar el correo.

En el servidor Web, existen muy pocas diferencias entre las versiones de linea de comandos de GPG y PGP, por lo que podemos utilizar GPG ya que resulta gratuito. En el equipo d~ el que leer el correo, puede que p,efiera utilizar una versión comercial de PGP por el complemento de interfaz gnifica que incorpora al lector de correo electrdnico.

Si no tiene un par de claves, genc!n!las en el equipo en el que vaya a leer el correo electrónico. Recuerde que un par de claves se compone de una clave pública que otros usuarios (asf como su secuencia de comandos de PHP) utilizarán para encriptar el correo antes de enviárnoslo y una clave privada que se utilizará para descifrar los mensajes recibidos o firmar el correo saliente. Es importante que el proceso de generación de claves se realice en el equipo en el que se va a leer el correo y no en el servidor Web ya que la clave privada no debe almacenarse en el servidor Web.

Si está utilizando la versión de lfnea de comandos de GPG para generar las claves, introduzca el siguiente comando:

```
gpg --gen-key
```

Deberá responder a una serie de preguntas. La mayor parte de ellas incluyen una respuesta predeterminada que puede aceptar. Además, se le pedirá su nombre real,

EL_HOLISTICO@HOTMAIL.COM - 2006

una dirección de correo electrónico y un comentario, que se utilizará para designar a la clave. Por ejemplo. 'Luke Welling clukeetangledweb.com.au>' .Como puede apreciar el patrón resulta claro. Si se incluyera un comentario, se colocaría entre el nombre y la dirección.

Para exportar la clave pública del par de claves, puede utilizar el comando:

```
9P9 ··•xport > nCllbredearebJvo
```

Como resultado se generará un archivo binario para su importación en el archivo de claves de GPG o PGP en otro equipo. Si quiere enviar por correo electrónico esta clave a otros usuarios, para que puedan importarla en aus propios archivos de claves, puede crear una versión ASCD como la siguiente:

```
gpg ··export -a> n0111Dredearebivo
```

Tras extraer la clave pública, puede cargar el archivo en su cuenta en el servidor Web. Para ello, puede utilizar FTP.

Los siguientes comandos asumen que estamos utilizando UNIX. Los pasos son los mismos que para Windows, pero los nombres de los directorios y los comandos del sistema serán diferentes.

Regístrese en el servidor con su cuenta y cambie los permisos sobre el archivo paria que los usuarios puedan leerlo. Escriba:

```
'ehlll0d 644 nolllbredearebivo
```

~ecesitará crear un archivo de ci~.;es para que el usuario ~tilizad~ para ejecutar las ~uencias de comandos de PHP pueda usar GPG. Este usuario dependerá de !=9.mQ esté configurado el servidor. A menudo suele ser el usuario nobody, pero puede ser cualquier otro.

Cambie al usuario del servidor Web. Necesitará disponer de acceso de usuario raíz para ello. En la mayor parte de los sistemas, el servidor Web se ejecuta como nobody. En los siguientes ejemplos partiremos de este supuesto. (Puede cambiarlo por el usuario adecuado en su sistema.) Si este fuera en caso en su sistema, escriba:

```
•ll root
a"  nobody
```

Cree un directorio para el usuario nobody y almacene el archivo de claves y otra información de configuración. Este archivo debería incluirse en el directorio raíz del usuario nobody. El directorio raíz de cada usuario se especifica en /etc/paaawd. En muchos sistemas Linux, el directorio del usuario nobody es el directorio predeterminado /; sobre el que dicho usuario no tiene permiso para esaibir. En muchos sistemas BSD, el directorio predeterminado para eJ usuario nobody es /noexistente; que como no existe, no se puede esaibir en él. En nuestro sistema, el usuario nobody se ha asignado al directorio raíz / tmp. Deberemos aseguramos de que el usuario de servidor Web dispone de un directorio raíz sobre el que tenga permiso para escribir.

Escriba

```
ed -
lllcdir .gnupg
```

El usuario nobody necesitará una clave de firma propia. Para crearla, ejecute el siguiente comando de nuevo:

```
9P9 •·gen- key
```

Como el usuario nobody es probable que no reciba ningún correo personal, podemos crear una clave de sólo firma. El único objetivo de esta clave es permitirnos confiar en la clave pública extraída anteriormente. Para importar la clave pública exportada anteriormente, utilice el siguiente código:

```
9P9 ··import noabredearebivo
```

Para indicarle a GPG que queremos confiar en esta clave, necesitamos editar sus propiedades de la siguiente forma:

```
gpg ··edit·key 'Luke lfelling cl11k..tangledveb.com.al&>'
```

En esta línea, el texto entre comillas es el nombre de la clave. Obviamente, el nombre de la clave no será 'Luke Welling cluke•tangledweb.com.au> •, sino una combinación del nombre, el comentario y la dirección de correo electrónico suministratda al generarlo.

Este programa incluye la opción help, que describirá los comandos disponibles trust, sign y save. Escriba trust para indicarle a GPG que conffa en la clave completamente. Escriba sign para firmar esta clave pública µtilizando la clave privada del usuarionóbody: Finalmente, utilice save para salir del programa manteniendo los cambios.

Probar GPG

Ya tenemos configurado GPG y listo para su uso. La creación de un archivo que contenga alguna secuencia de texto y su almacenamiento como test .txt nos permitirá utilizarlo para probarlo.

Escriba el siguiente comando (modificado para utilizar el nombre de su clave).

```
gpg ·• ··recipieAt 'Luk• Welling cllkettaagled~eb.com.al&>' ··enerypt teat.txt
```

debería devolver la siguiente advertencia

```
gpg: Warning: using insecure memory!
```

y crear un archivo llamado test .txt. ase. Si abre este archivo, debería ver un mensaje encriptado como el siguiente:

```
·····BIGI»   fGP HSSSIG1····-
Ver.ton, GnufG v1.0.3   lGIIU/LiAuxl
C01111ent: For info aee http:// .....gnupg.org
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
hQIIOAODU7b'VQgdtaSAQA!lr4RgR7xpIBaXICillLQw8S+k1QdQ+p/PsqL8tICrQ+Bl
OG.JTBeblUDBrwqUv/uQLTda0rloPSrIAZ7c,GVtthOYSVBj211111kT81IIBvdo~50yll
Xll'UCv9/rLlrJlkxe4Vp8QPBTSBlPdII/ly8VPSgST87gAga0811PflSllPqwlly'rkD
/2oJBVL,elcPl84a0181r8llDllOUAAhc:jjXt2Dlt/llXlq,P18QIIS,utc:uon4DPaW10
/pllCkcYDgLcXfBjbU/'l'Clllllpa To7kX4CicIb'l'IClIIIHY4UdaCWQf271oS+l.il
cJltSCIIIIIPioI,NNllllCQRY,plbfxL2u839IRJrQba,xo..0nkBOuTYxlLOTG•PrBrB..
tvBVNIOnaHU7RJay+oY41833pkS+IIIIIVWYuaJwlvR:tdlXZIIV,wx4~2XGTl7b28V
WSBDWOo8111ZaPYkQlUITOq,sBlxP8y+YJv8B3x4pxdROxa+!lpqbB7q3+XXlldlOhDR
+u7UNxTLDbgC+llll
•gfQu
·····SIID POP NSIIIAGB·····
```

Debería poder transferir esle archivo al sistema en que gener6 la clave inicialmente y ejecutar

```
gpg teat.txt.aac
```

para ver el texto original de nuevo. El texto se escribir' en un archivo con el mismo nombre que tenía antes, en este caso test. txt.

Para que el texto se reproduzca en pantalla, utiliza el indicador -d:

```
gpg ·d teat.txt ... c
```

Para colocar el texto en el archivo que desee, en lugar de utilizar el nombre predeterminado, puede utilizar el modificador -e y especificar un archivo de salida <t, la siguiente forma:

```
9'11 -do teat.out teat.txt.aac
```

Si tiene configurado GPG para que el usuario utilizado para ejecutar las secuencias de comandos PHP pueda usarlo desde la linea de comandos, pdcticamente ha terminado. Si no fundonara, consulte a su administrador del sistema o la documentadón de GPG.

Los listados 17.1 y 17.2 permiten enviar correo electr6nlco encriptado utilizando PHP para llamar a GPG.

Llatado 17.1. prtvate_mall.php. Nuestro formularlä HTML .,.,. envlllr corrao eleclr6nlco encrlpmdo.

```
cbtal>
cbody>
cbl>Send Me Pr1vata Mailc/bl>

c?pbp
   // puede que necaaite caabiar ••ta linea, ai ao utilixa
   //loa puertoa pll:9deter1linadoa1 80 para el tr&f1co normal y 443 para SSL
   1f( SDVD.('IDVD. PORT'| l•Ull
      ecbo •cp..cfont color•"red">
            WAaJIIIIII: you llave not c-cted to tbia page uaing SSL.
            Your •••ag• could ba nad by otbera.c/font>c/p>'1
```

return $this->$name;


```
Tour ...11 addnaa,cbr />
ciaput type.•taxt• Dallll•"froa• •l••···>a•>cbr ,..
Subject,cbr />
cinput type••text• na... •title• •l••••Ja•..cbr ,..
Your -••age:cbr />
ctextarea Da118••body• col•••Jo• rowa-•10•>
c/taxtana>cbr />
clnput typa••IIUbalt• valua•"Sendl •.,
c/fOrll>
c/body>
c/btab
```

Umdo 17.2. Nnd..JNW*_mel.php. Sec:uencla de comandos de PHP ,_. u.m. a GPG y .,... canw electr6illcos encriptlldoa.

```
c?pbp
   //cree II0llbrea de varlabl•• corto•
   $fromi • $_POST(•fromi•J1
   $title • $_P0ST('titla•J,
   $body • $_POST('body')1

   $to___ il • 'lllkaelocalbNt• 1

   // Indiqua a gpg dAlade encoatrır el archivo da clave•  -,.....
   // Bn ••t• aiate.., el directorio ralx del uauario nobody ••'llllll/
   putanv( 'GIIUPOBONa•/t:19/ .~•  11

   //cree uis noabre de udli1N1 acluaivo
   $lnflla • t-...naa(•,  •pgp•J1
   $outfila • $lnflla.•.aaC"'1

   //eacriba •l taxto del usuario en el archivo
   $fp • fapen($infile,  •w'1
   fwritel$fp,  $body11
   fcloae($fp11

   //configura al CCla8lldo
   $C0llallDd • •/uar/local/bin/gpg -a\\
              ··reclpient 'Luke ..lling clukeetangledweb.coa.au>'  \\
              ··•ncrypt ·o $outfil• $lnfila•1

   // •:t•cuta al cOIIUldo gpg
   ayat..($comlll8Dd, $reault11

   //elimina •l archivo temporal no ancriptado
   uislinkl$lnfile11

   $fp • fcipen($outflle, 'r•J¦
   1fl1$fp!(flleaixe  ($outfilel••OI
   (
      Snault • -11

   •|••
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
        //lu  el  archivo eacriptado
        scoacenca • frud  1$fp. fileaise  ISoutfilell;
        //elimine el archivo temporal encriptado
        unlinkl$outfilel;

        ..111sto  ...11.  $title, $coateata,  'Froa,  Sfroa\a•);
        echo  'chl>Neaa.,. Seacc/bl>
                cp>Your ..aaage waa  encrypted and aent.c/p>
                cp>Tbank you.•c/p>';


    if(SreaulCl•OI
    (
       echo  'chl>Brror,c/bl>
          cp>Your ..aaage could not be encrypted, ao baa not been aent.c/p>
          cp>Sorry.c/p>';
```

Para que este código funcione, necesitará cambiar algunos elementos. El correo electrónico se enviará a la dirección incluida en $to_email.

En el listado 17.2 tendrá que modificar la linea

```
p~cen;('CIIUPGIIONB•/tmp/.gnupg'I;
```

para.~ la nbicadóo del arclwro de claves GPG. En nues.tm sistema. el servidor Web se ejecutará como usuario nobody y el directorio raíz será /tmp/.

Estamos utilizando la función t111pna111( ) para crear un nombre de archivo temporal exclusivo, Puede especificar el directorio y un prefijo de nombre. de archivo. Vamos a crear y a eliminar estos archivos en un segundo, por lo que no resulta muy importante cómo los llamemos. Vamos a especificar el prefijo 'pgp', pero vamos a permitir que PHP utilice el directorio temporal del usuario.

La instrucción

```
$ca1111and •   •/uar/local/bin/gpg ·a  •.
        ··recjpieat  ·Luke  Nelling cluk..tangledweb.c.,,...au>'   '
        '··eacrypt  -o  $outfile  $1nfile';
```

configura el comando y los parámetros que se utilizarán para llamar a GPG. Deberá modificarlo para adaptarlo a sus necesidades. Al utilizarlo en la línea de comandos tendrá que indicar a GPG qué clave utilizar para encriptar el mensaje.

La instrucción

```
system($command, $result);
```

ejecuta las instrucciones almacenadas en $command y almacena el valor devuelto en $re.sult. Podemos ignorar el valor devuelto, pero nos permite tener una instruc~ón i f e indicar al usuario que algo ha ido mal.

Cuando hayamos terminado con los archivos temporales utilizados, los eliminamos con la función unl ink ( ). Por lo tanto, el correo no encriptado del usuario se

almacenará muy poco tiempo en el servidor. Y si el servidor fallara durante la ejecución, el archivo podría quedarse en el servidor.

Al considerar la seguridad de nuestra secuencia de comandos, es importante tener en cuenta todos los flujos de información dentro del sistema. GPG encriplllrá nuestra secuencia de comandos y permitiñ que nuestro destinatario la descifre, pero cabe preguntarse cómo llegó la información originalmente desde el remitente. Si suministramos una interfaz Web para enviar correos GPG enaiptados, el flujo de la informaci~n seguiría la ruta ilustrada en la figura 17.5.

Figura 17.5. En nuestra aplicación de cono elec:trónico cifrado, el mensaje se envla a trav6s de lntemet trea veces.

En esta figura, cada flecha representa el mensaje enviado desde un equipo a otro. Cada vez que se envía el mensaje, viaja a través de Internet y puede pasar por una gran cantidad de redes y equipos intermedios.

La secuencia de comandos que estamos examinando aquí existe en el equipo denominado Servidor Web en el diagrama. En el servidor Web, el mensaje se. encriptará utilizando la clave pública del destinatario. A continuación se enviará~ través de SMPTal senidor de cc,rreo del destinatario. &te se eeneefllft a s115el'Yl-dor de correo eiectrónico, utilizando POP o lMAP probablemente, y descargará el mensaje utilizando un lector de correo. En este punto descifraremos el mensaje utilizando su clave privada.

Los datos transferidos en la figura 17.5 se etiquetan como 1, 2 y 3. Para las fases 2 y 3, la información transmitida es ~ mensaje GPG encriptado y n~ tienen nin~ valor para nadie que no tenga la clave privada. Para la transferencia l, el m~Je transmitido es el texto que el remitente introduce en el formulario.

Si nuestra información es lo suficientemente importante como para encriptarla en la segunda y tercera etapa del viaje, es un poco tonto enviarla sin encriptar en la primera fase. Por lo tanto, esta secuencia de comandos pertenece a un servidor que utiliza SSL

Si intentamos conectamos a nuestra secuencia de comandos sin SSL, se generará una advertencia. Puede comprobarlo por medio del valor $_SERVER 'SBRVER_ PORT• J . Las conexiones SSL proceden del puerto 443. Cualquier otra conexión provocará un error.

En lugar de suministrarun mensaje de error, podemos procesar esta situación de otras formas. Podemos redirigir al usuario al mismo URL a través de una conexión SSL Podemos optar por ignorarlo porque por regla general no resulta importantesi el formulario se entrega utilizando una conexión segura. Lo que si suele ser importante es que los detalles que el usuario ha introducido en el formulario se nos envien de forma segura. Bastaría con indicar un URL completo como acción en nuestro formulario.

EL_HOLISTICO@HOTMAIL.COM - 2006

En la actualidad, la etiqueta de apertura del formulario pneenta este aspecto:

```
<form method="post" action="send_private_mail.php">
```

Poc:bfamos modificarla para enviar datos a1ravá de SSL incluso si el usuario se conecta sin SSL,. de la liguiente forma!

```
<form method="post" action="https://webserver/send_private_mail.php">
```

Si codificamos de manera espedfica el URL completo de esta forma, nos uegu· ruemos de que los datos del visitante se envfen utiliado SSL. pero necesitaremoe modificar el c6digo cada vez que lo utilicemos en otro lerYidor o incluso en otro directorio.

Aunque en este cuo y en muchos otros, no es importante enviar el fonnulario al usuario a travfs de SSL, resulta IIIIII buena idea hacerlo IIIf. Si aparece el súnbolo de IU\ candado en la barra de estado de IIIIII navegadores, los usuarios podrút estar seguros de que la informac:l6n se esbl enviando de forma segura. No necesitará examinar el código H'IML para detenninar el atribulo de aa:i6n.

Lecturas adicionales

L.l.~ de la ven1m 3 de SSLpodñ encontrarla en la Web de Nellleape: `bttp://boale.netacape.ca11/eng/aa13.`

Sf. desea saber mú IOlne cómo funcionan las ndes y los protocolos de redes, puede consultar IU\ texto de introducci6n ya clúico eom,uttr N,hllorb de .Andrew S. Tanenbaum.

A continuación

Con esto terminamos el an6liaia de los upectos relacionados mn el comercio e1ectnSnico y la seguridad. En la siguiente secd6n, examinuemos licnicu mú avanzadas de PHP, como la interac:d6n con otros equipos en Internet, la generación de ilúgenes de forma instantmea y el uso del control de sesión.

EL_HOLISTICO@HOTMAIL.COM - 2006

Parte IV

Técnicas avanzadas de PHP

con al C JV o~ y el servidor

En un capíttdo anterior, vimos cómo leer y escribir datos en archivos del servidor Web. En este capítulo, vamos a examinar otras funciones de PHP que nos permiten interactuar con el sistema de archivos en el servidor Web. En este capitulo analizaremos los siguientes aspectos:

- Cargar archivos con PHP
- Utilizar funciones de directorio
- Interactuar con archivos en el servidor
- Ejecutar programas en el servidor
- Utilizar variables de entorno de servidor

Para analizar los usos de estas funciones, examinaremos un ejemplo. Suponga que queremos permitir que nuestros clientes puedan actualizar parte de los contenidos de un sitio Web, por ejemplo, las noticias sobre nuestra compañía. (O que queremos incorporar una interfaz más sencilla que la de FTP o SCP para su uso.) Una posibilidad podría consistir en permitir a los clientes que carguen archivos de contenido en forma de texto sin procesar. Estos archivos estarían disponibles.en el sitio, a través de una plantilla diseñada con PHP, como hicimos en un capitulo anterior.

Antes de profundizar en las funciones del sistema de archivos, vamos a examinar brevemente el fundónamiento de la carga de archivos.

Introducción a la carga de archivos

PHP incorpora una funcionalidad de gran utilidad que permite realizar cargas HTI'P. En lugar de recorrer el camino del sérvidor al navegador utilizando HTI'P, los archivos realizan el trayecto contrario, es decir, van desde el navegador al servidor. Por regla general, esta función se implementa mediante una interfaz de formulario HTML. En la figura 18.1 se ilustra la que utilizaremos en nuestro ejemplo:

sFIgura 18.1. El fonnulario HTML utilizado para la carga de este an:hlvo consta de campos y tipos de campo diferentes a los Incluidos en un formulario HMTL normal.

Como puede ver, el formulario incluye un cuadro en el que el usuario puede introducir un nombre de archivo o hacer clic sobre el botón Browee... para examinar los archivos disponibles localmente. Es posible que no haya visto un formulario decarga antes. En un instante veremos cómo implementarlo.

Tras introducir un nombre de archivo, el usuario puede hacer clic sobre Smd Pile para cargar el archivo en el servidor, donde le espera una secuencia de comandos de PHP.

HTML para la carga de archivos

Pata poder implementar la carga de archivos, necesitamos utilizar sintaxis de HTML disponible para dicho fin. En el listado 18.1 se recoge el código HMTL correspondiente a este formulario.

Listado 11.1. upload.html. Formularlo HTML para la carga de archivos.

```
cht.lftl>
chead>
   ct.itle>Adllini•t.rat.ion  • upload aew fileec/t.it.le>
c/head>
cbody>
chl>Upload new new• fil•••/hl>
cfora enct.yp••"•ult.ipart./fora-dat.a• act.ion••upload.php• ..t.hod•"po•t.">
   cinput. t.yp••"bidden• n•..•"IIAJ:°FILB SIZB" valu••"1000000">
   Upload t.hi• file: cinput. na...-;uerfile• t.ype••Ule">
   cinput. t.yp• ••aublllit.• value••Send rile">
```



```
e/fon>
c/body>
c/bt.111>
```

.. .Pfjele en que este formulario utiliza POST. Las cargas de ——vo tambn se pueden realizar con el método PUT admitido-por Netscape Composer y Amaya. No funcionan con GBT. Las funciones adicionales de este formulario son las siguientes:

- En la etiqueta cfom>, debemos establecer el atributo enctype••aultipart/form-data• para permitir que el Nffidor 1epa que viene un archivo con la infonnad6n habitual del formulario.
- Debemos incluir un campo de formulario que establezca el archivo de tamafto máximo que se puede cargar. Se trata de un campo oculto y se muestra a continuación:

  ```
  cinput. t.yp••"hidden• ne...••NAX_FJLII_SJIB" value•"1000000">
  ```

  El nombre de este campo de formulario debe ser MAX_PILB_SIZB. El valor es el tamafto mximo (en bytes) de archivos que permitiremos que se carguen. Por el momento lo hemos establecido en 1000000 bytes (apie191madamente un megabyte). Puede aumentar o disminuir este tamai\o en función de su aplicación.
- Necesitamos una entrada de tipo file, que en el ejemplo se corresponden con la línea

  ```
  <input name="userfile" type="file">
  ```

  Puede seleccionar el nombre que desee para el archivo, pero téngalo en cuenta cuando vaya a utilizarlo para acceder a su archivo desde la secuencia de comandos PHP de recepción.

Un inciso sobre seguridad

Antes de seguir adelante, conviene recordar que algunas versiones de PHP presentan ciertos puntos débiles en el código para la carga de archivos. Si decide utilizar la funcwn de carga de archivos en un servidor de producción, es aconsejable que utilice la versión mú actual de PHP y que se mantenga atento a los posibles parches que puedan surgir. Este hecho no debería disuadirle de utilizar una tecnología tan buena. Basta con poner atención al escribir el código y considerar la posibilidad de restringir el acceso a la función de carga de archivos a los administradores del sitio y a los encargados de administrar el contenido, por ejemplo.

Código PHP para procesar la tarea de carga del archivo

La tarea de escribir PHP para capturar el archivo resulta bastante sencilla, pero depende de la versión de PHP y de los parAmetros de configuración. Los nombres

EL_HOLISTICO@HOTMAIL.COM - 2006

de funciones y Yariables han cambiado con respecto a versiones anteriores y dependen de si ha activado register_globals o no. El código de nuestro ejemplo no necesita que se active esta directiva, pero sí requiere una versión de PHP posterior a la 4.1.

Al cargar un archivo, se incluirá en una ubicación temporal en el servidor Web. Se trata del directorio temporal predeterminado del servidor Web. Si no mueve o cambia el nombre del archivo antes de que termine la ejecución de la secuencia de comandm, se eliminar&.

Los datos que procesará la secuencia de comandos de PHP se almacenan en la matriz superglobal $_FILES. Si ha activado el parámetn>.register_globals, puede acceder a la información a travá de nombres de variable directos. Sin embargo, &te es el ámbito en el que resulta más importante mantener desactivado el parámetroregister_globals o al menos actuar como si lo estuviese y utilizar la matriz superglobal e ignorar las variables globales.

Lu entradas de $_FILES se almacenan con el nombre de la etiqueta <file> del formulario HTML El elemento de formulario tiene el nombre userfile, por lo que la matriz incluirá los siguientes contenidos:

- Elvaforalmacenadoen$_FILES['userfile•J [•tmp_name•J esellugar en el que el archivo se ha almacenado temporalmente en el servidor Web.
- El valoralmacenadoen$_FILES ['userfile• J [•name •J es el nombre del archivo en el sistema del usuario.
- El valoralmacenado en $ FILES [•userf ile •J [' si ze •J es el tamailo del archivo en bytes.
- El valoralmacenado en $_FILES [•userfile' l C •type !J es el tipo MIME del archivo, por ejemplo text/plaino i-ge/gif.
- El valor almacenado en$ FILE['userfile•J [•error•] devolverá cualquier código de error asociado con la carga de archivos. Se ailadi6 en PHP 42.0.

Como sabemos dónde se encuentra el archivo y cómo se llama, podemos copiarlo en cualquier ubicación que nos resulte útiL Al final de la ejecud6n de la secuencia de comandos, se eliminará el archivo temporal. Por lo tanto, debe trasladarlo de ubicación o asignarle otro nombre si desea guardarlo.

En nuestro ejemplo, vamos a utilizar los archivos cargados como artículos de noticias recientes para eliminar todas las etiquetas que puede incluir y a trasladarlo a un directorio más útil. En el listado 18.2 se recoge una secuencia de comandos que realiza esta operación.

Umdo 11.2. upload.php. PHP para captulW'schlvos dnde el formulario HTML.

```
clatal>
cllead>
   ctitle>Vploadiag''' c/title>
c/--4>
cbody>
```



```
chl>IJpla.ding file •..c/hl>
c?php

  if (S_PILBS(•u•erfile'J ['error•) > 0I
  {
    echo 'Probl••• ';
    •witch (S_u•erfil• _error)
    {
      cen 1: echo 'Pile u:ceected upload_MX_file•i:r.e•; breek,
      cen 2: echo 'Pile exceeded MX file •ize•: br••k
      ca•• 3: echo 'Pil• only partially uploadad': break;
      caae 4: echo 'llo fil• uploaded'; break,-

    exit1


  // ¿Ll•va aaignado el archivo el tipo NIIIS correcto?
  if (S_PILBS[•uaerfileJ ('type•J I• •tu:t/plaia•)
  {
    echo 'Probl••= file ia aot plain text•;
    exit;


  // coloqu• el archivo donde d••••
  $upfil• • '/uploada/' .$_PILES [•uaerfile•) c ——— J

  (f lia_uploadad_file1$_PILES[•u•erfile•) [•tlllll_n...•JII

    ttTI116ve_usiroaa.a:n1.-u:_ntu fiü•itrf Uii•ffit¡.jp"_iwüt";T; supr ú~, ,
    {
      echo 'Problem: Could not move file to daatinatioa direceory•;
      exit;


  elae
  {
    echo 'Probl••• Poa•ible file upload attack. Pilen...,
    echo $_PILBS(•uaerfile•J (•na .. •);
    exit;


  echo 'Pile uploaded •ucce••fullycbr>cbr>'¡

  // volver a aplicar formato a lo• contenido• dal archivo
  $fp • fopen($upflle, •r•);
  $contenta• fread 1$fp, file•ize 1$upfile1)
  fclon 1$fp1:

  $conteata • atrip_tag•1$content•I;
  Sfp • fopenC$upfile, •v•I;
  fvrite1$fp, $contenta);
  fcloae($fp1;

  // —t.rar qu6 •• ha cergado
  echo 'Previ— of uploaded file c—t•nt•:cbr />chr />'¡
  echo Sconteata,
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
    ecbo  'cbr  />chr  />'l

?>
c/body>
c/html>
```

Uama la atención que la mayor parte del código de esta secuencia de comandos esté dirigido a la comprobación de errores. La carga de archivos implica riesgos potenciales de seguridad que deben evitarse siempre que resulte posible. Necesitamos validar el archivo cargado con mucho cuidado para aseguramos de que resulta seguro distribuirlo a nuestros visitantes.

Vamos a examinar las partes fundamentales de esta secuencia de comandos. Comencemos por la comprobación del código de error devuelto por $_PILBS (•userf ile •l (•error• l . Este código de error se introdujo en PHP 4.2.0. Desde PHP 4.3 también existe una constante asociada a cada uno de estos códigos. Las constantes *y* fos valores posibles son los siguientes:

- UPLOAD_BRROR_OI(,, valor 0, significa que no se produjo ningún error.
- UPLOAD_BRR_INI_SIZB, valor 1, significa que el tamatlo del archivo cargado supera el máximo valor especificado en el archivo php . ini con la directiva upload_max_filesize.
- UPLOAD_BRR_PORM_SIZBvalor 2, significa que el tamatlo del archivo cargado supera el valor mAximo especificadq en el formulario HTML del elemento MAX_PILB_SIZB.
- UPLOAD_BRR_PARTIALvalor 3, significa que el archivo se ha cargado parcialmente.
- UPLOAD_BRR_NO_FILB, valor 4, significa que no se ha cargado ningún archivo.

Si desea utilizar una versión antigua de PHP, puede crear una versión manual de estas comprobacionespor medio del código de ejemplo del manual de PHP o de ediciones anteriores de este libro.

También podemos comprobar el tipo MIME. En este caso, sólo-queremos cargar archivos de texto, por lo que probamos el tipo MIME asegurándonos de que $_FILES (•userfile •) ('type• J contiene text/plain. Realmente se trata de comprobación de errores, no de seguridad. El navegador Web del usuario determina el tipo MIME a partir de la extensión del archivo pasado al servidor. Si el hecho de pasar uno falso sirviera para algo, un usuario malintencionado no tendría problemas para hacerlo.

Seguidamente, comprobamos que el archivo que estamos intentando abrir se ha cargado y no se trata de un archivo local como /!!!tc/pasawd. Volveremos sobre este tema en un instante. Si todo esto funciona, copiamos el archivo en nuestro directorio de inclusión. En este ejemplo, hemos utilizado /uploads/, que se encuentra fuera del árbol de documentos Web y resulta, por lo tanto, un buen lugar en que almacenar los archivos que deban dirigirse a otras ubicaciones.

A continuación, abrimos el archivo, limpiamos las etiquetas de HTML o PHP que puedan haber quedado COI\ la función strip_tag• () *y* volvemos a guardarlo. Por último, mostramos los contenidos del archivo para que el usuario pueda ver que se ha cargado satisfactoriamente.

En la figura 18.2 se muestran los res.ultad0l! de la ejeglció.n(S!!tisfactoria) de esta secuencia de comandos.

Figura 11.2. Tru copiar et archivo y aplicarle formato, se muestra el.g.lvo cargado pera Indicar el usuario que le operación de carga H hll realizado coffitclamente.

En septiembre de 2000, se anunció que un pirata infonmlico podfa hacer que la secuencia de comandos para la carga de archivos procesara un archivo local como si se hubiera cargado en el servidor. Este hecho se documentó en la lista de correo BUGTRAQ. Si lo desea, puede consultar los documentos sobre seguridad oficiales en uno de los muchos compendios BUGTRAQ como http: //lista. insecure. org/bugtraq/2000/Sep/0237.html.

En nuestro caso hemos utilizado las funciones is_uploaded_file () *y* move_uploaded_file () para aseguramos de que el archivo que estamos procesando se ha cargado *y* que no se trata de un archivo local como /etc/paSBwd. Esta función está disponible a partir de la versión 4.0.3 de PHP.

Si no se escribe con cuidado la secuencia de comandos pera procesar la carga de archivos, un visitante malintencionado podría utilizar un nombre de archivo temporal *y* convencer a la secuencia de comandos pare que procese dicho archivo como si se tratara de un archivo cargado. Como muchas secuencias de comandos pare la carga de archivos devuelven los datos cargados al usuario o los almacenan en algún lugar desde los que puedan cargarse, se podría lograr acceder a cualquier archivo que el servidor Web pueda leer, como los archivos del directorio /etc/paSBwd *y* código fuente de PHP, incluidas las contraseilas de la base de datos.

Problemas habituales

Al realizar cargas de archivo en un servidor es necesario tener en cuenta una serie de aspectos.

EL_HOLISTICO@HOTMAIL.COM - 2006

- En el e;emplo anterior se asume que los usuarios han sido autenticados en algún momento. No es aconsejable permitir que todo el muruio pueda cargar archivos en un sitio.
- Si va a permitir que los usuarios no autenticados o no fiables carguen archivos en el servidor, conviene examinar exhaustivamente su contenido para evitar que se cargue o ejecute una secuencia de comandos malintencionada. Debería estar atento, no sólo al tipo y a los contenidos del archivo, sino tambifn al nombre del archivo en sí mismo. Es aconsejable cambiar el nombre a los archivos cargados utilizando instancias que resulten seguras.
- Si está utilizando un equipo basado en Windows, asegmese de que utilizar \ \ o / en lugar de \ en las Ntas de archivo.
- El uso del nombre de archivo proporcionado por el usuario, como hemos hecho en esta secuencia de comandos, puede generar diversos problemas. El más evidente es que se corre el riesgo de sobrescribir accidentalmente archivos existentes si alguien carga un archivo que tenga el nombre de otro ya utilizado. Un riesgo menos evidente es que diferentes sistemas operativos e incluso diferentes configuraciones de idioma local permiten diferentes conjuntos de caracteres válidos para los nombres de archivo. Un archivo cargado puede tener un nombre con caracteres que no sean válidos en nuestro sistema.
- St etproceso de-cargano fwicim.a; comptuebe el aiddvopbp. ini. l/erifi't1H que la directiva upload_tmp_dir apunta a algún directorio al que disponga de acceso. Puede que también necesite ajustar la directiva memory_l imit para cargar an:hiv()!!de mayor tamai\o. Esta directiva determina el tamaAo máximo de los archivos que puede cargar. En Apache también se incluyen tiempos de espera configurables y limites de tamafto de transacciones que debe comprobar si tiene problemas para cargar archivos de gran tamaAo.

Utilizar las funciones de directorio

Una vez cargados los archivos, resultará útil permitir ver lo que se ha cargado y manipular los archivos de contenido. PHP consta de una serie de funciones de sistema de archivos y directorios que resultan de utilidad al respecto.

Leer desde directorios

En ppmer lugar, implementaremos una secuencia de comandos que permita examinar el contenido cargado. La operadón de examinar directorios resulta muy sencilla en PHP. En el listado 18.3, se incluye una sencilla secuencia de cmnandos que se puede utilizar con este fin.

Llat.do 11.3. bNMMrdir.php. Un listado de dll'9Ct0rias de los archivas cargados.

```
cbtal>
cbeacb
   ctitla>Brov .. Diractoria•c/titla>
c/baacb
cbody>
cbl>Brov•ingc/bl>
c?pbp
   $c:urrant_dir  •  •/upload•/';
   $dir • opandir($current_dirl;

   ec:bo •cp>Upload diractory l• $currant_dirc/p>"1
   acbo •cp>Diractory Li•ting:c/p>cUl>';
   vbila ($fila•   raaddir($dir)I
   (
      ec:bo •cli>$filac/li>";

   acbo 'c/ul>';
   cl0Hdir($dirl1
?>
c/body>
c/btab
```

Esta secuencia de comandos utiliza las funciones opendi r (l, closedi r <l y readdir().

La f\mción opendir O se utiliza para abrir un directorio para su lectura. Su uso es muy similar al de fopen (l para leer desde archivos. En lugar de pasarle un nombre de an:hivo, debería pasarle un nombre de directorio:

```
$dir •opandir($currant.dirl1
```

La función devuelve un indicador de directorio, de la misma forma que f open (l devuelve un indicador de archivo.

Cuando el directorio está abierto, puede leer un nombre de archivo desde él Uamancloa readdir ($dirl, como se muestra en el ejemplo. Este elemento devuelve false cuando no hay archivos que leer. (Tenga en cuenta que también devolverá fal8e si lee un archivo llamado •o•; conviene verificarlos si es probable que ocurra.) Los archivos no se ordenan en ningún orden concreto, por lo que si necesita una lista ordenada, debería leerlos en una matriz y ordenarlos después.

Cuando haya terminado de leer desde un directorio, deberá llamar a closedir ($dirl para terminar. De nuevo, la operación es similar a fcloae (l para un archivo. El resultado de ejemplo de la secuencia de comandos de exploración de directorios se muestra en la figura 18.3.

Si está utilizando este mecanismo para permitir que el usuario examine los diredorios, es aconsejable limitar los dinctorias que se pueden explorar para evitar el acceso a áreas normalmente restringidas.

Una función asociada que resulta de utilidad en ocasiones a la función rewinddi r ($dir), Esta función restablece la lectura de los nombres de archivo al principio del directorio.

EL_HOLISTICO@HOTMAIL.COM - 2006

Como alternativa a estas funciones, puede utilizar la clase dir de PHP. !sta consta de las propiedades handle y path. y de los métodos read (), el ose () y rewind ( ), que funcionan igual que las alternativas no incluidas en clases.

Figura 18.3. El listado de directorio muestra todos loa archivos del directorio seleccionado, Incluidos el directorio. (directorio actual) y el directorio .• (el directorio del nivel superior). Puede filtrarlos al lo desea.

Obtener información sobre el directorio actual

~ Podemos obtener información adicional utilizando una ruta a un archivo.

Las funciones dirname($ruta) y basename ($ruta) devuelven la parte de una ruta y la parte de un archivo de nombre de la ruta, respectivamente. Esto podría resultar útil para nuestro explorador de directorios, en especial si comenzamos a construir una estructura compleja de directorios de contenido basada en nombres de directorios y nombres de archivos descriptivos.

También podemos agregar al listado de directorios una indicación del espado libre para cargas de archivos con la función disk_free_apace ($ruta). Si pasamos una ruta a un directorio, devolveri el número 4le bytes libres en el disco (Windows) o el sistema de archivos (Unix) en el que se encuentra él directorio.

Crear y eliminar directorios

Además de leer de manera pasiva la información sobre directorios, podemos utilizar las funciones mkdi r ( ) y rmdir () para crear y eliminar directorios. Sólo p~rá crear y eliminar directorios en rutas a las que disponga de acceso el usuario utilizado para ejecutar la secuencia de comandos. El uso de la función mkdir () resulta Ns complicado de lo que podría parecer a primera vista. Esta función toma dos parámetros, la ruta al directorio deseado (incluyendo el nuevo nombre de directorio) y el permiso que le gustaría que tuviera dicho directorio, por ejemplo:

```
llkdir(•/t11p/t••tin9•, 07771:
```


Sin embargo, los permisos que se enumeren no tienen por qué ser los permisos que vaya a obtener. Se combina úna versión inversa de la función umaak actual con este valor, por medio de AND, para obtener los permisos actuales. Por ejemplo, si la función umaak es 022. obtendremos el permiso 0755.

Si loclesea puede restabt~umaak antes de crear un directorio para sustituir este efecto mediante el siguiente código:

```
$oldulli. .lt • -11:(01 ,
llkdirl"/t.,/t•ting•, 077711
1111a•ltlSoldlulaakll
```

Este código utiliza la función umaak (›, que se puede utilizar para comprobar y cambiar la wuak actual. Se sustituir, la U1Uak por el valor pasado y se devolveri la umask antigua o, si se llama sin parmetros, se devolver, únicamente la uuak actual. Tenga en cuenta que la función umask () no tiene ningún efecto en los sistemas Windows. La función rmdi r ( ) elimina un directorio. Por ejemplo:

```
:nidir(•/t.,/t•ting"I;
```

o

```
:nidir(•c:\\t;sP\\te•ting"l1
```

El directorio que se desee eliminar debe estar vado.

Interactuar con el sistema de archivos

Ademú de ver y obtener información sobre directorios, podemos interactuar y obtener información sobre archivos incluidos en el servidor Web. Ya hemos visto cómo escribir y leer archivos, pero existen muchas otras funciones de archivo disponibles.

Obtener información sobre archivos

Podemos alterar la parte de la secuenáa de comandos para explorar directorios que leen archivos de la siguiente forma:

```
wbile |$file•  •dir->readlll
{
   echo •ca bref•°filedetail•.pbp?file••.$file.'">'.$file.•c/a>cbr>';
```

Podemos crear la secuencia de comandos f i ledeta i la . php para suministrar información adicional sobre un archivo. En el listado 18.4se muestran los contenidos de este archivo. Tenga en cuenta que parte de las funciones utilizadas aquí no son compatibles con Windows, entre las que se incluyen pos ix_getpwuid O, fileowner () y filegroup (), o su compatibilidad no es fiable.

EL_HOLISTICO@HOTMAIL.COM - 2006

Llaudo 11.4. flledetalla.php. Funciones de NtIldo de 8n:hlvo y sus .....a.dos.

```
cht•l,.
chead,.
   ctitle,.Pile Detail•c/title,.
c/lwad,.
cbody,.
c?php
   Scurreot_dir • '/uploed•/';
   Sfile·  ba•eaa..(Sfilel;
   //filtrarla  informaci6n de directorio por  •eguridad

   ecbo 'cbl.,.Detaila  of fila:  '.Sfil•.'c/hl,.';
   Sfila • $currant_dir.$filel

   acbo 'ch2,.Pila  datac/b2,.•;
   acbo 'Pila  la•t  acc••••d:  '.daca(•j   F Y R:i•,  filaati..(Sfilal).'cbr    j,.•;
   ecbo 'Fila laet modified:  '.data(•j   F Y R:i•,  fil..ti..(Sfila)l.'cbr   j,.•;

   Sueer • poeix_getpwuid(fileownar($filel);
   acbo 'Pila ownar, '.$uear[•aa .. •I. 'cbr  j,.•,

   S91N>UP • poeix_getgrvid(filegroup($filall)
   acbo 'Pila group, '.$group[•aa—•)  .•cbr  j,.•;

   acbo  'Pila parmi••ione:  '.decoct(filaparme(SfilalJ.'cbr    j,.,,

   actw  •rila  type,  '.!Uaeypelffil•l  ....._.  f+,L,-

   •cr}lo  'Fila  •ise:  •.fila•isa(Sfilal.'   byteecbr  j,.,,

   acllo  'ch2,.File  ta•tec/h2,.'

   acbo  •ie dir:  '.(le   dir($filal7   'true•    'falea•).'chr    j,.•;
   acbo  'ie:exacutable7  '.(ie_axecutabla(Sfilal?    •trua•  :  'felea•l.~br    j,.,,
   acbo  'i•  fila:  '.(i•   file($filel?   •true•  :  •falea'l.'cbr   ,,..,
   ecbo  •ie-link:  •.(ie-link(Sfile17   •trua•  :  •fal..•).'cbr  j,.•,
   acbo  •ie-readable:  ~(i•  raadable(Sfilal? •true•   •falea•).'cbr   j,.•;
   acbo  'ie:writable:  '.(ie:writabla(Sfilel?  •true•  :  'falea•).'cbr   j,.,,

7,.
c/body,.
c/ht•l,.
```

En la figura 18.4 se muestran los resultados de ejecutar el listado 18.4.

Vamos a analizar qué hace cada una de las funciones utilizada en el listado 18.4. Como se mencionó anteriormente, la función bHen-• () obtiene el nombre del ardüvo SM el directorio. (Tambiál puede utilizar la función dima- () para obtener el nombre del directorio SM el nombre de archivo.)

Las funciones fileatime () y fil-time() devuelven la marca de tiempo indicando el momento en el que se accedió por última vez al ardüvo y la última vez que se modificó, respectivamente. Hemos modificado el formato de la marca de tiempo con la función date () para que resulte mú legible. Estas funciones devol-

verán ~l mwno valor en algunos sistemas operativoe (como en el ejemplo) según qué información almacene el sistema•

Figura 11.4. La vista de loa delalles de en:hlvo muestra lnbmaci6n del sistema de archivos sobr9 un archivo. Como puede oburYar los permisos se muntnt en formato octal.

Las funciones fi leowner () y filegroup (-) devuelven el Id. del usuario (uid) y el Id. de grupo (gid) del archivo. astos se pueden convertir en nombres utilizando las funciones posix_getpwuid () y poaix_getgrgid () para que resulten más sencillos de leer. Estas funciones toman el uid o gid como parámetro y devuelven una mabiz asociativa de informaciál sobre el usuario o grupo, incluido sus nombres, como hemos hecho en esta secuencia de comandos.

La función fileperms () devuelve los permisos sobre el archivo. Les hemos aplicado un nuevo formato utilizando la función decoct () para que resulte más familiar a los usuarios de UNIX.

La fundón filetype () devuelve información sobre el tipo de archivo que se va a examinar. Los resultados posibles son fifo, char, dir, block, link, file y UDknown. La función filesize () devuelve el tamafto del archivo en bytes.

El segundo conjunto defunciones(ia_dir(), ia_executable (), ia_file (), is_link(), is_readable () e ia_writable ()) prueba el atributo con nombre de un archivo y devuelve true o falae.

Tambiin podrfamos utilizar la fundón atat () para recopilar una gran caiatidad de infonnac:ión del mismo tipo. Al puu un alddvo, devuelve una matriz que contiene datos similares a los de estas funciones. La funci6n i.tat () es similar pero para su uso con "vínculossimbólicos.

EL_HOLISTICO@HOTMAIL.COM - 2006

Las funciones,de estado de archivo tardan mucho tiempo en procesarse. Por esta razón, sus resultados se almacenan en caché. Si desea comprobar información sobre el archivo antes o después de un cambio, deberá llamar a

clearatatcacheCJl

par:a J>C?Cler vaciar los resultados anteriores. Si desea utilizar la secuencia de comandos previa antes o después de cambiar datos del archivo, deberla llamar a esta función como primer paso para asegurarse de que los datos generados estúl actualizados.

Cambiar propiedades de archivo

Además de ver las propiedades de archivo, podemos alterarlas. Las funciones chgrp(archivo, grupo), chmod(archivo, permiso) y chown(archivo, usuario) se comportan de forma similar a sus equivalentes de UNIX. Ninguna de ellas funcionará en los sistemas basados en Window,, aunque chown ( ) se ejecuta-n y devolverá siempre true.

La función chgrp (l se utiliza para cambiar el grupo de un archivo. Sólo se puede utilizar para cambiar el grupo al que pertenece el usuario, a menos que se trate del usuario raiz.

La función chmod (l se utiliza para cambiar los permisos sobre un archivo. Los pe.alisos ae paan con la forma cmaod habitual de UNIX (debe anteponerles un •o• para indicar que su formato es octal, por ejemplo).

chmodC•aomefile.txt•, 077711

La función chown (l se utiliza para cambiar el titular de un archivo. Sólo se puede utilizar si la secuencia de comandos se está ejecutando como raiz, lo que no deberla ocurrir nunca.

Crear, eliminar ·l desolazar archivos

Puede utilizar las funciones de sistema de archivos para crear, trasladar y eliminar archivos. En primer lugar, y de forma sencilla, puede crear un archivo o cambiar la hora de la última modificación utilizando la función touch (). Su funcionamiento es similar al comando touch de UNIX. Su sintaxis es la siguiente:

int touch Catring *archivo,* lint hora!, iat hora))>

Si el archivo ya existiese, su hora de modificación se sustituiría por la actual o ~o~ la indi.cada en el segundo parámetro si se especifica. Si desea utilizar esta ul~~ opción, deberla incluirse en formato de marca de tiempo. Si el archivo no ex1Stiesese cre.iria. La hora de acceso del archivo también se modificarla: de manera predeterminada la hora del sistema actual o la marca de tiempo especificada en parámetro opcional. .

Puede eliminar archivoe utilizando la función unlink ( ). (Tenga en cuenta que esta función no se llama delete: no existe dicha función.) Se utiliza de la siguiente forma:

ualiakCSna..bre_de_archivol1

Esta función no resulta operativa con las versiones antiguas de Windows. Si éste fuera su caso, puede utilizar la siguiente función para eliminar un archivo en Windows:

ayatu("del aa..bre_de_arcbivo.ext•) l

Puede copiar y mover archivos con las funciones copy (l y rename (l de la siguiente forma:

copy(S.ruta_origea, Sruta_deatinol /

renaM (Sarch1vo_ant1guo, $arch1w_nuevol l

Como puede haber observado, hemos utilizado la funddn copy () en el listado 18.2. La función rename <l se encarga tambWn de la tarea de trasladar archivos de un lugar a otro porque PHP no consta de la función move. La pos~d de mover archivos entre sistemas de archivos y la de sobrescribir archivéi1con rename () depende del sistema operativo, por lo que debe comprobar los et'=tos en su servidor. Asf mismo, tenga cuidado con la ruta que utilizan hasta el nombre de archivo. Si fuera relativa, lo seria con respecto a la ubicación de la secuencia de comandos, no al archivo original.

Utilizar funciones de ejecución de programas

A continua~ón, dejaremos a un lado las funciones reladonadas con el sistema de archivos y pasaremos a examinar las funciones disponibles para ejecutar comandos en el servidor.

Estas funciones resultan útiles cuando se quiere suministrar una interfáz de usuario basada en la Web a un sistema existente basado en línea de comandos. Por ejemplo, hemos utilizado estos comandos para crear una interfaz de comandos para el administrador de listas de correos ezmlm. Volveremos a utilizarlas en capftulos posteriores.

Existen cuatro tá:nicas principales para ejecutar un comando en el servidor Web. Aunque son bastante parecidas, presentan pequeftas diferencias.

1. exec (J : La función exec (l tiene la siguiente sintaxis:

   atriag exec l•triag ~ndo l, array reaultado l, int valor.:_devu•l to) l l

   Se le pasa el comando que deseamos ejecutar, por ejemplo,

   exec(•la -la"I;

EL_HOLISTICO@HOTMAIL.COM - 2006

La fundón exec () no genera un resultado directo. Devuelve la última lfnea del resultado del comando.

Si se pasa en una variable como resultado, se obtendrá un matriz de cadenas que representa cada línea del resultado. Si se pasa en una variable como valor_devuelto, se obtendrá el código de devolución.

2. paHthru ‹l: La función paHthru (l tiene la siguiente sintaxis:

```
void pa••tbru C•tring e0lm9Jldo l, inc veJor_devueJtoJI
```

La función p&Htbru () repite el resultado en el navegador, lo cual resulta útil si el resultado es binario (por ejemplo, dal.OS de una imagen). No devuelve nada.

Los parámetros funcionan de la misma forma que los de la función exec ().

3. system (): La función system () tiene la siguiente sintaxis:

```
ecring •Y•t•• (ecring e — do l, inc veJor_devuel:oJl
```

Esta función repite el resultado del comando en el navegador. Se diferencia de la funci9n anterior en que intenta eliminar el resultado tras cada línea (si se ejecuta PHP como módulo de servidor). Devuelve la última línea de resultado (si tiene éxito) o false (en caso de fallo).

Sus parámetros funcionan de la misma forma que los parámetros de las tunctonH:anteriUles.

4. Apdstrom invertidos: Se trata de operadores de ejecución como ya se indicó en un capítulo anterior.

No tienen un resultado directo. El resultado de ejecutar el comando se devuelve en una cadena, que se puede dirigir a un dispositivo o utilizar para realizar cualquier otra operación deseada.

Si sus necesidades resultan m4s complicadas, puede utilizar popen (l, proc_open () y proc_cloae (). Estas funciones se utilizan para bifurcar procesos externos y enviar dal.OS entre ellos. Las últimas dos funciones se agregaron en la versión 4.3 de PHP.

La secuencia de comandos que se recoge en el listado 18.5 ilustra cómo utilizar las cuatro t-=nicas de forma equivalente.

Uatado 11.S. pn,gex.php. Funciones de estado de archivo y sus resultacloe.

```
c?pbp

   ebdirl'/uploede/'lr

/////  ,,_.:1611 exec
   eclo •cpm>';

   //  1111:lx
```



```
   exee(•l• -1•', $reev.ltl:
   JI windawl
   JI exee('dir', $reeultl;
   foreaeb ($re•ulc •• $linel
     eebo o$line\n•,

   •ello • c/pn»•;
   eebo •cbr />cbr /><br />';

///// ver•i611 pe••tbru
   ecbo •cpre> ';

   JI unix
   pa••tbru('l• -1•'l;
   JI wind-
   // paHtbru('dir' I t

   eebo 'e/pre>' ;
   edlo 'cbr />cbr />cbr />';

///// ver•i611 •Y•t..

   echo 'cpre>' 1
   JI unix
   $re•ult • •Y•t••<'l• -1•'l;
   JI windoH
   JI Sr•eulc • •Y•t••l'dir'I;
   ACIM:l....~L:..
   9C:bo •cbr />cbr />cbr />';

/////Yer•i611 eon ap6•crofe• inYertido•

   ecbo •cpre>!
   // unix
   Sr••ulc • '1. -el ';
   // windoH
   JI Sre•ulc • "dir ';
   eebo $re•ult;
   echo •e/pre>• :
```

Podrwnos haber utilizado esl.OS enfoques como alternativa a la secuencia de comandos de exploración de directorios escrita anteriormente. Uno de los efectos secundarios del uso de funciones externas queda ampliamente demostrado aquí: su código dejar4 de ser portable.

En este caso hemos utilizado comandos de Unix y el código no se ejecutan. en Windows. Si tiene previsto incluir dal.OS enviados por el usuario como parte del comando que va a ejecutar, deberla ejecutarlos siempre a través de la función eacapeshellcmd () en primer lugar. De esta forma evitará que los usuarios puedan ejecutar comandos malintencionados en su equipo. Puede llamar a esta función de la siguiente forma;

```
•Y.t••Ce.eepe•bellemd(Sc:omllUICIII;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Tambiél deberfa utilizar esta func:i6n para convertir cualquier argumento especial que tenga previsto pasar a su comando de nl lc:leo.

Interactuar con el entorno: getenv() y putenv()

Antes de dar por concluida esta sección, vamos a ver cómo poclemoe utilizar variables de en~ desde PHP. Para ello, 6ponemos de dos funciones: getenv (›, que permite recuperar variables de entorno y putenv (› que permite establecer variables de entorno. Tenga en cuenta que estamos hablando del entorno en el que PHP 1e ejecuta en el servidor.

Puede obtener una lista de todas las variables de entorno de PHP ejecutando pbpinfo (› • Algunas resultan mú útiles que otras, por ejemplo

Listado 6.5. reflection.php. Mostrar información sobre la clase Page.

devolved el URL de la f"gina desde la que el usuario lleg6 bata la f"gina actual. Tambiénpuedeestablecerlasvariablesdeentornocomodeseeconputenv(I, por ejemplo,

```
$ShoM • •t——/m,body•,
put-Y (• IICNII•$~ 0) 1
```

∴srfuera mi iiclmúiistraaordesiiiteriiasy qüislera Dmítit ras viíiitíla de entomCJ 4Uepueden establecer los programadores,podría utilizar la directiva Hf e_mode_allowed_env_vara de pbp. ini. Cuando PHP 1e ejecuta en modo aeguro, los usuarios s6lo pueden establecer las variables de entorno cuyoe prefijoe 1e enumeran en esta directiva. Si desea obtener información adicional sobre quf representan las variables de entorno, puede consultar la espedficad6n de CGI, que enconlrar6 enbttP.://booboo.ncaa.uiuc.ec!u/cgi/env.btml.

lecturas adi~ionales

La mayor parte de las funciones de sistemas de archivo de PHP est6 asociada a las funciones subyacentes de los sistemas operativos. Si utiliza UNIX y desea obtener más información, consulte las P'ginas 11a11.

A continuación

En el siguiente capitulo, veremos c:6mo utilizar las funciones de sed y protocolo de PHP para interactuar con sistemas distintos a los de nuestro servidor Web, lo cual contribuid a ampliar el homonte de posibilidades para nuestras secuencias de comandos.

EL_HOLISTICO@HOTMAIL.COM - 2006

19

Utilizar funciones de red y de protocoJo

En este capítulo, examinaremos las funciones orientadas a la red de PHP que permiten a nuestras secuencias de comandos interactuar con el resto de Internet. En Internet, encontrará una ingente cantidad de recursos y una gran variedad de protocolos para su uso. En esta sección examinaremos los siguientes aspectos;

- Descripción general de los protocolos disponibles
- Enviar y leer correo electrónico
- Utilizar otros sitios Web a través de HTTP
- Utilizar funciones de búsqueda de red
- Utilizar FTP

Descripción general de los protocolos disponibles

Los protocolos son reglas de comunicación para una situación dada. Por ejemplo, ya conocemos el protocolo que hay que seguir cuando nos encontramos a alguien. Decimos hola, estrechamos la mano y decimos adiós. En diferentes situaciones se requieren diferentes protocolos. Por otra parte, gente de diferentes culturas esperan diferentes protocolos, lo que puede dificultar la interacción. Los

EL_HOLISTICO@HOTMAIL.COM - 2006

protocolos de red son similares. Como en el caso de los protocolos humanos, se utilizan diferentes protocolos informáticos para cada situl.ción y aplicaciones. Por ejemplo, utilizamos HTIP (Protocolo de transferencia de hipertexto) para enviar y recibir páginas Web. Es probable .que también haya..utili.zado FTP (Protocolo de transferencia de archivos) para transferir archivos entre equipos de una red. Existen muchos otros.

Los protocolos, así como otros estándares de Internet, se describen en las RFC (Solicitudes de comentarios). Estos protocolos son definidos por el IETF (Grupo de trabajo de ingeniería de Internet). En Internet existen muchos lugares en los que encontrar RFC. La fuente básica es el editor de RCF situado en http://www. rf e-editor. org.

Si tiene problemas al trabajar con un protocolo dado, las RFC son la fuente oficial y suelen resultar útiles para solucionar los problemas con el código. Sin embargo, la cantidad de detalles es muy grande y suelen incluir cientos de páginas.

Entre las RFC mú conocidas se puede citar la RFC2616,que describe el protocolo HTIP/1.1 y la RFC822,que describe el formato de los mensajes de correo electrónico por Internet.

En este capítulo, vamos a examinar los aspectos de PHP que utilizan algunos protocolos. Específicamente, veremos cómo enviar correo con SMTP, cómo leer correo con POP e IMAP, cómo conectar con otros servidores Web a través de HTIP y HTIPS, y cómo transferir archivos con FTP.

Epviar y recibir correos electrónicos

La forma principal de enviar correos electrónico en PHP consiste en utilizar la función mail (), ya comentada en un capítulo anterior. Esta función utiliza el protocolo SMTP (Protocolo simple para la transferencia de correo) para enviar correo electrónico.

Es posible utilizar toda una variedad de clases gratuitas para aftadir funcionalidad a mail (). En un capítulo posterior, utilizaremos una 8-ase compl'11\entaria para enviar archivos adjuntos HTML en un mensaje de correo. SMTP sólo se utiliza para enviar correos. Los protocolos IMAP (Protocolo de acceso a mensajes de Internet, descrito en la RFC2060) y POP (Protocolo de oficina de correos, descrito en la RFC1939o STD0053) se utilizan para leer el correo de un servidor de correos. Estos protocolos no pueden enviar correo.

IMAP se utiliza para leer y manipular mensajes de correo almacenados en un servidor y es más avanzado que POP, el cual se suele utilizar para d!!SCargamensajes de correo electrónico a un cliente y eliminarlos del servidor.

PHP incorpora una biblioteca IMAP. Esta biblioteca también se puede utilizar para realizar conexiones POP, NNTP (Protocolo de transferencia de noticias de red) elMAP.

Examinaremos con detenimiento el uso de la biblioteca IMAP en el proyecto incluido en un capítulo posteri~r.

Utilizar otros sitios Web

La Web permite utilizar, modificar e incrustar servicios e infonnaci6n existente en nuestru propias pjginas. PHP convierte esta tarea en algo muy sencillo, como se ilustra en el ejemplo que se comenta a continuación.

Imagine que la compaftfa para la que trabaja quiere mostrar en la página principal el precio de sus acciones. Esta información está disporu'ble en varios sitios dedicados a temas bursátiles, pero ¿cómo acceder a ella?

Tenemos que empezar por buscar un URL en el que se incluya esta información de primera mano. Tras ello, cuando un usuario se dirija a nuestra página principal, abriremos una conexión a dicho URL para recuperar la página y extraer la información deseada.

Como ejemplo, hemos creado una secuencia de comandos que recupera y aplica formato a la cotización extraída del sitio Web de AMEX. En concreto, recuperaremos el precio de cotización de las acciones de Amazon. (Puede incluir otra información ya que el principio es el mismo.) En el listado 19.1 se recoge la secuencia de comandos.

u.taclo 19.1. lookup.php. Secuencia de comandos que recupera la cotlziélón de una acción del NASDAQ con el slmbolo indicado en Ssymbol. ~

```
chtal>
chead>
   ctitle>Stock Quote fr0II RASDIIQc/title>
c/head>
cbody>
c?php
   // aeleccione qut acción exatlinar
   $ayabol•'ANZR';
   ecbo •chl>Stock Quote for $ayabolc/hl>"

   $theurl•"http://W11W.aaex.coa/eqÜitiea/liatCllp•
            ,•/BqLCDetQuote.jap?Product_Syabol•$ayabol•;

      if (!($contenta• file_get_contenta($theurl)))

      echo 'COUld not open URL';
      exit;


   // buaque la parte de la p4gina deaeada y devutlvala como reeultado
   $pattern • • (\\\$Co-, J+\\. 10-,1 +I •;

   if (eregi($pat~ern, $contenta, $quotell
   (
      echo "$ayabol vaa la~t aold at: •
      echo $quote[11 1
      echo •c/p>'

   elae
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
    {
       echo •cp>HO quoce availablec/p>';
    };

    // reconozca la fuence
    echo ‹cp>
       .'Tbia informacion recrieved from cbr />'
       .•ca bref•\"$Cheurl\">$Cbeurlc/a>cbr />"
       ' •oa '. |dace('l jS F Y g:i a T'II;
?>
c/body>
c/bUll>
```

En la figura 19.1 se recoge el resultado del listado 19.1.

Figura 11.1. Esta secuencia de comandos utiliza una expresión regular para extraer la a,tización de una acción de la información recuperada de la bolsa de valores.

La sea,iencia de comandos resulta bastante clara. De hecho, no utiliza ninguna fundón nueva, sólo nuevas aplicado~ de dichas funciones.

Puede que recuerde de un capítulo anterior que podemos utilizar las funciones de.archivo para leer desde un URL. Eso es lo que hemos hecho en este caso. La llamada a, file_get_contents ()

```
Scoacenc• • file_gec_concenca($Cbeuzl)
```

devuelve todo el texto de la página Web a la que apunta el URL almacenado en $contenta.

Las funciones de archivo sirven para mucho en PHP. En este ejemplo simplemente se carga una página Web a través de HTI'P, pero podrfamos interactuar con otros servidores por medio de HTIPS, FTP u otros protocolos de la misma forma. Para determinadas tareas puede que necesite adoptar un enfoque más especializado. Parte de la funcionalidad de PHP se encuentra en funciones de FTP conaetas y no está disponible a través de fopen () u otras funciones de archivo. En un apartado posterior del capítulo encontrará un ejemplo de uso de funciones de FIP. Para algunas tareas HTTP o HTIPS puede que tenga que utilizar la biblioteca ~L-itL,que le permite iniciar sesión en un sitio Web e imitar el progreso de un usuano a través de algunas páginas.

Una vez obtenido el texto de la página Web, podemos utilizarlo como expresión regular y recurrir a la función eregi (l para buscar la parte de la página deseada:

```
Spaccern • •(\\\$(0·9 J+\\.(0·91+)•;
if |eregi($paccern, $conceaca, $quoce))
{
   ecbo "$aymbol vae laac aold ac: •:
   ecbo Squoce(11 :
   echo •c/p>';
```

Y eso es todo. Podemos utilizar esta tmüca parar realizar otras tareas. Por ejemplo, podemos recuperar la información meteorológica local e incluirla en nuestra página. El mejor uso de esta técnica consiste en combinar información procedente de varias fuentes para agregarle algún valor. Un buen ejemplo es la secuencia de comandos creada por Philip Greenspun que genera la página Bill Gates Wealth Clock:http://philip.greenspun.com/WealthCloclt.

Esta página toma la información de dos fuentes. Obtiene la población actual de los EEUU de la oficina del censo estadounidense. A continuación, examina el valor de una acción de Microsoft y combina los dos datos, agrega una dosis saludable de la opinión del autor y genera nueva información: el valor aproximado de Bill Gates.

Si va a utilizar una fuente de información externa como áta con fines comerciales, conviene comprobar antes si puede hacerlo ya que en algunos casos es necesario tener en cuenta aspectos relativos a la propiedad intelectual.

Si est4 aeando una. secuencia de comandoe E0111e ésta, es probeble qae desee pasarle algunos datos. Por ejemplo, si está conectándose a un URL externo, es probable que desee pasar algunos parámetros escritos por el usuario. Si áte fuera el caso, conviene utilizar la función url~mc;:~_() . Esta función tema una cadena y la convierte al formato correcto para un URL. Por ejemplo, tránsforma los espacios en signos más. Puede llamarla de la siguiente manera:

```
Seacodedpara..ter • urleacode($para.cerl;
```

Este enfoque presenta un problema: que el sitio del que se obtiene la información varíe el formato de los datos, lo que impediría el funcionamiento de nuestra secuencia de comandos. Una alternativa a este sistema son los servicios Web. Son como objetos remotos a los que podemos conectamos para recuperar datos como cotizaciones bursátiles. La compatibilidad de PHP con los servicios Web aumenta día a día. Además de ventajas técnicas, el uso de servidos Web implica la utilización de una interfaz y de un dispositivo a los que el proveedor nos permite acceder, ya sea de forma implícita o explícita, por medio de un programa.

Utilizar funciones de búsqueda de red

PHP incorpora un cpnjunto de funciones de "búsqueda" que se pueden utilizar para comprobar información sobre nombres de host, direcciones IP e intercambios

EL_HOLISTICO@HOTMAIL.COM - 2006

de correo elec:trónico. Por ejemplo, si estii aeando un directorio como Yahoo!, cuando se remitan nuevas direcciones podría comprobar automáticamente que el host de un URL y la información de contacto de dicho sitio es v41ida.De esta forma püede contribuir a reducir la labor posterioMle los revisores al examinar el sttio si descubren que no existe o que la dirección de co.rreo electrónico no es v61ida.

El listado 19.2 muestra el c6digo HTML correspondiente a un formulario de envío para un directorio como éste.

Llat.do 19.2. dlrec:tory_aubmlt.html. Código HTML p.-. el formui.to de envio.

```
chtalJo
chead>
   ctitle..Subait your aitec/title..
c/bead..
cbodyJ0
chl..Subllit aitec/blJo
cfora ..thod•"poat• actiOD•"directory_aubllit.php•..
UJIL: cinput type••text• DaINt•°url• ais..•Jo• value.•bttp1//• ..cbr ¡.,
Bmail contact: cinput type••text• na.e•• ... 11• aise•"23"Jo<br ¡.,
cinput type•"•ub•it• na..•"Submit aite•..
c/fora..
c/body..
c/ht•b
```

Se trata de un formulario bastante sencillo. En la figura 19.2se recoge la versión resultante, con datos de ejemplo introducidos.

Figura 19.2. El envio de dlrec:clonH a un directorio 1uele n,querlr la lncluli6n de un URL y detalln de conr.c:to pa111 que lol adminlltredorn del dinlclorlo puedan aviumoa del momenlo en el que M agrega nue,llll dlrecci6n al directorio.

Al pulsar el botón de envfo, en primer lugar tenemos que comprobar que el URL está alojado en un equipo real y, despu&, que la parte del host de la dirección de correo electrónico también ~tii en un equipo real. Hemos escrito una secuencia de comandos para comprobar estos elementos y en la figura 19.3 se recoge el resultado. La secuencia de comandos encargada de realizar estas comprobaciones utiliza dos funciones del conjunto de funciones de red de PHP: gethostbyname(l y dns_get_mx(l. En el listado 19.3 se incluye el c6digo de la secuencia completa.

Figura 19.3. Ella venl6n de la MCUenC:la de camanc1o1 fflUHlra las '911111tadoa de comp,obar IOI nombrel de hOlt de lol URL y dlnlccionea de correo elec:lr6nlco. e, probable que um veni6n de proclucc:l6n no muntre estoa rnulmdol, pera • irlllreunte vw la lnformecl6n devuella por lal camprobaclana.

USUdo 19.3. dlnlclary_aubmit.php.~ de camanclol p.-. vert1 1c8r el URL y la dlreccl6tl de correo electr6nlco.

```
cht•l..
chead>
   ctitle..site aublli••ion reaultac/titleJ0
c/head..
cbodyJ0
chlJoSite aublia•ion re•ultac/blJo
c?pbp

   // Extraiga l'0a campo• del formulario

   $url • $ JmQUBST(•url'11
   $e..11 .-$_RBQUBST(•...11•J1

   // Coaprueba el URL

   $url • parae_url1$url);
   $hoat • $url(•boat•J;
   ifl1($1p • gethoatbyname1$boat111
   (
      ecbo 'Boat for URL doa• not bave valid iP•1
      exit;


   echo •eoat i• at IP $1p cbr ¡,.•,

   // COllpruebe la direcc16n de correo electrdnico

   $ — 11 • axplodel•e•, $...u,,
   $e..ilhoat • $e..11(1);

   // fl:IHe en que la func16n dDa_get__ O •no .. i•pl—ta• ~
   // laa veraionea de Willdowa de PBP
   U 1!daa_get_-1$—Uboat, $mxboataarr1>
   (
     echo 'lmail addre•• i• not at valid hoat•1
```

```
    exit;

echo •s.ail i• delivered via,
foreach |$mcboetearr ae $11XI
   ecbo•su •;

// Si ee abre, todo perfecto

echo 'cbr />All eublntted detaila are ok.cbr />';
echo 'Tbaak you for eut.ittillg yaur ei.te.cbr /:,,'
   .'It will be viaited by oae of our etaff ..-bar• eoon.•

// Ba ua caao real, agregue a la baee de dato• de eitioa en eepera •.'

c/body>
c/htal>
```

Vamos a examinar las partes interesantes de esta secuencia de comandos.

En primer lugar, tomamos el URL y le aplicamos la función pane _url (l • Esta función devuelve una matriz asociativa de las diferentes partes de un URL Las partes disponibles de la información son scheme, user, pass, host, port, path. query y fragment. Por regla general, no vamos a necesitar todas ellas, pero a continuación se incluye un ejemplo de cómo forman un URL

Dado un URL como http: / /nobocly: secret•bigcompany.com: 80/ script. php?variable•value#anchor los valores de cada una de las partes de la matriz ,eráw.-

- scheme: http://
- user: nobody
- pass: secret
- host: bigcompany.com
- port: 80
- path: script.php
- query: variable=value
- fragment: anchor

En nuestra secuencia de comandos, sólo queremos información sobre hos~ por lo que la extraemos de la siguiente forma:

```
$url • par .. url($url)1
$boat • Surl('boet•J,
```

Tras ello, podemQSobtener la dirección IP del host, si se encuentra en el DNS,. Para eUq•podemos utilizar la función gethostbyname(l , que devuelve la dirección IP si la hubiera o false en ca_so contrario:

```
$ip • gecboatbyna .. ($boetl;
```


También puede seguir el otro camino utilizando la función gethos tbyaddr (l, que toma una IP como parámetro y devuelve el nombre de host. Si llama a estas funciones de manera sucesiva, podría obtener un nombre de host distinto al utilizado al principio, lo que implica que el sitio está utilizando un servicio de host virtual. Si el URL es válido, podemos comprobar la dirección de correo electrónico. En primer lugar la dividimos en nombre de usuario y nombre de host con una llamada a explode e) :

```
$e—u • explode l.,. , $.-111;
$e..ilboet • $e..il|11;
```

Tras obtener la parte del host de la dirección, podemos comprobar si existe un lugar al que enviar dicho correo utilizando la función dns_get_mx(l:

```
dne_get_mc(Se ..ilhoet. $11Xboetaarrl;
```

Esta función devuelve un conjunto de registros MX (del inglá Mail Exchange, Intercambio de correo) para una dirección de la matriz suministrada en $mxhostarr.

Los registros MX se almacenan en el DNS y se buscan como un nombre de host. El equipo indicado en el registro MX no tiene por qué ser necesaria~ente el equipo en el que acabará finalmente el correo. En su lugar podría tratarse de un equipo que sepa dónde dirigir el correo. (Puede haber más de uno, en cuyo caso la función devolverá una matriz en lugar de una cadena con el nombre de usuario.) Si no tiene un registro MX en el DNS, el correo no tendrá dónde dirigirse.

Tenga en cuenta que la fund6ndns_get_mx(J no seüñp"1enierifaenluversiones de Windows de PHP. Si todas estas comprobaciones resultan correctas, podemos colocar los datos del formulario en una base de datos para su posterior revisión por un miembro del personal,

Además de las funciones que acabamos de utilizar, puede utilizar la función genérica checkdnsrr (l, que toma un nombre de host y devuelve true si incluye algún registro en el DNS.

Utilizar FTP

El Protocolo de transferencia de archivos. o FrP, se utiliza para transferir archivos entre host de una red. En PHP, puede utilizar la función fopen <l así como varias funciones de archivo con FTP también disponibles con conexiones HTIP, para conectar y transferir archivos desde y hacia un servidor FrP. Sin embargo, también se incluye un conjunto de funciones específicas de FrP que vienen con la instalación PHP estútdar.

Estas funciones no se incluyen en la instalación estándar de manera predeterminada. Para poder utilizarlas en UNIX, es necesario ejecutar el programa conf igure de PHPcon la opción - -enable-ftp y volveraejecutarmake. Si va a utilizar la instalación estándar de Windows, las funciones de FrP se habilitan de manera automática.

EL_HOLISTICO@HOTMAIL.COM - 2006

Utilizar FTP para realizar una copia o reproducir un archivo

Lu funciones de Fl'P lOn litiles para mover o copiar loe archivos desde o hasta otros host. Se suelen utilizar para hacer copias de eeguridad del sitio Web o para reproducir archivos situados en otra ubicación. Examinaremoe un lel\dllo ejemplQ que utiliza funciones Fl'P para reproducir un archivo. En el listado 19.4 se incluye la secuencia de comandos correspondiente a este ejemplo.

Llemdo 11.4. ftpmlrror.php. Secuencia de comandos para delcargar nuevas versiones de un archivo dNde un sen,ldar FTP.

```
cbtal>
cuad>
   ctitle>Nirror updatec/title>
c/uad>
cllody>
<l\l>Nirror updatec/bl>
c?pbp
//configure varialll••• aodiflquela• para aju•tarl•• a •u aplicaci6a
$bo•t .. •ftp.c•.rait.edu.au•,
$uaer • ·-on.,.ou•'1
$pa•aword • •••1xaaple.c-·;
$rjllll0tefile • '/pub/t•g/teratera/tt••bl4.sip•,
Sl!><=alUle· • '1t111P1WdtllblWlttHhll .lip<"T.

//~oon6ctHe al boat
$conn • ftp_connect(•$bo•t•)1
if (1$CODD)
(
   ecbo 'Brror: Could DOt connect to ftp ••rvercbr />'1
   -it¡
}
ecbo •connected to $bo•t.cbr />"1

11 inicie •••i6a en el bo•t
e $reault • ftp_login($conn, $uaer, $pa••>1
if (1$re•ult)
(
   ecbo •Brror: Could not log OD u Suaarcbr />";
   ftp_quit($CODD)¡
   -it¡
}
ecbo "Logged in a• $u•ercbr />";

11 coapruebe la inforaaci6n de fecha de lo• archivo• para detarainar
// •i nec••ita actualizarlo•
ecbo 'Cbecking file tiN••• cbr />'1
if (fil•--i•t•1$localfile))
(
   $localti.. • fil..ti..($localfilel1
   ecbo 'Local file la•t updated '1
```



```
   echo date('G:i j·N·Y', $localti..);
   ecbo •cbr />'1
}
el••
   $localti••O;
$-tati•.   ftp_adta($conn, $reaotefile)1
if (1 ($-teti.. - 0))
(
    II Bato no aignifica qua el arcbiw DO ••t• abl1 puede que el Hn·i..ior
    // DO adaita el tinpo de aodificaci6n
   ecbo •can\•t acceH -t•   file tiM.cbr />'1
   s-t•tiN•$localti•+l1//     a•egdze•• de qua axi•t• una actualisaci.:-:t
}
•]••
(
   echo 'llaaote fila laat updated '1
   ecbo date(•G,i j-N·Y', s-t•ti•l1
   acbo •cbr />';
}
if (1($-Ceti .. ,. $localti..ll
(
    acbo •Local c:opy ia up to date.cbr />'1
    axit1

II deecargua el arcbiw
ecbo •Gatting fil• fzoa ••rver•••cbr />';
$fp • fopen {SlOcalfile,    •W•11
if (ISaucc••• • ftp_fgat($COIIII, Sfp, $rell0tafile, rTP_8:nrARY) 1..
(
   ecbo 'Brror: Could aot download file';
   ftp_quit($CODD);
   axit1
}
fclOH($fp);
acbo 'Pila downloaded •ucc•••fully';

// cierre la conex16a al bo•t
ftp_quit($c:oan);

;"
c/body>
c/btab
```

En la figura 19.4 se muestra el resultado de ejecutar esta secuencia de com,tndos una vez.

Se trata de una secuencia de comandos bastmte genérica. Comienza por '-'l't.1blecer algunas va~bles:

```
$bo•t • 'ftp.c•.rait.edu.au';
Suaar • 'anonyaou•'1
Spa•-d • ·~la.-•;
$-t•fila • '/pub/tag/teratera/tt••bl4·sip';
$localfila • •/tap/vritabla/tt••blt.sip•,
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Flgu,. 19.4. La HCUencia da comandos para al sitio FTP de nipllcas comprueba si la veraión local de un archivo está actualizada y descarga una nueva veraión en caso de que no lo est6.

La variable $hoat debería contener el nombre del servidor FTP al que desee conectarse y las variables $user y $password se correspondenal nombre de usuario y contraseÑl que le gustaría utilizar para iniciar la sesión. Muchos sitios FTP admiten lo que se conoce como sesiones anónimas en las que se puede utilizar un nombre de usuario puesto a disposición de todo el mundo para poder establecer la conexión. No es necesario utilizar ninguna contraserut,pero es habitual suministrar la dirección de correo electrónico en su lugar para que el administrador del sistema pueda saber de dónde proceden los usuarios. &ta es la convención que hemos seguido aquf.

La Yariable $ramo~afila contiene la Rita huta el al'dli·,e c¡ue CfUHeMesdes.-cargar. En este caso, vamos a descargar y reproducir una copia local del Tera Term SSH y el cliente SSH para Windows. (SSH equivale a "secure shell", núcleo seguro. Se trata de una forma encriptada de Telnet.)

La variable $localfile contiene la ruta a la ubicación en la que vamos a almacenar el archivo descargado en nuestro equipo. En este caso, h~os aeado un directorio llamado / tmp/ wr i table con permisos establecidos para que PHP pueda escribir un archivo. Independientemente de su sistema operativo, tendrá que aear este directorio para que funcione la secuencia de comandos. Si su sistema operativo cuenta con permisos estrictos, compruebe si permiten que la secuencia de comandos pueda escribir. Debería poder cambiar estas variables para adaptar la secuencia de comandos a sus necesidades.

lo$ puos básicos seguidos en esta secuencia de comandos son los mismos que si quisiéramos enviar por FTP el archivo desde una interfaz de linea de comandos:

1. Conectarse al servidor remoto de FTP.
2. Iniciar la sesión (bien como un usuario normal o como ail6nimo).
3- Comprobar si el archivo remoto se ha actualizado.
4 Si se ha actualizado, descargarlo.
5. Cerrar la conexión a FTP.

Vamos a examinar cada uno de estos pasos uno a uno.

Conectarse al servidor FTP remoto

Este paso equivale a esaibir

```
ftp bo8tnaM
```

en la línea de comandos en la plataforma Windows o en la plataforma UNIX. Este ~so se realiza en PHP con el siguiente c6_digo:

```
$conn • ftp_connect("$1108t"1:
U  (1$connl
(
   ecbo 'Error:  Could not connec:t to ftp urv•rcbr  />'1
   exit;
1
ec:110 •coanected to $bo•t.cbr  />";
```

La llamada a la función aquf es ftp_connect 1). Esta función toma un nombre de host como parámetro y devuelve un identificador a una conexión o false si la conexión no pudo establecerse. La función también puede tomar el número de puerto del host al que conectarse como segundo pará,netro opcional. (A~~ ~ hemos utilizado este parámetro.) Si no especifica un número de puel'lo, se utilizará el puerto 21 que es el puerto FTP predeterminado.

Iniciar sesión en el servidor FTP

El siguiente paso consiste en iniciar sesión como US\18rio dad.etton W\ contraseña concreta. Para ello, puede utilizar la función ftp_login O:

```
• $r••ult • ftp_login($conn,  $uur,  $pa••l1
if  (1$rHult1
(
   ecbo "Error:  Could not log oD •• $u••rcbr  />";
   ftp_quit($coanl,
   exit;
1
ecbo •i.ogged  in••  $u•ercbr  />";
```

La función toma tres parámetros: una conexión FTP (obtenida desde ftp_connect ()),un nombre de usuario y una contraserut. Devolverá true si el usuario puede iniciar la sesión y false si no puede hacerlo. Como observará hemos colocado un simbolo O al principio de la linea para eliminar errores. De esta forma, si el usuario no puede iniciar sesión obtendremos una advertencia de PHP en nuestra ventana del navegador. Podemos capturar el error como hemos hecho aquf probando la variable $result y suministrando un mensaje de error más explicativo. Tenga en cuenta que si el intento de inicio de sesión fallara, se cerraría la conexión FTP utilizando ftp_quit (1 (en un instan~ ampliaremos este punto).

Comprobar el tiempo de actualización de archivos

Como estamos actualizando una copia local de un archivo, resulta lógico com~ probar primero si es necesario actualizar el archivo ya que no queremos volver a

EL_HOLISTICO@HOTMAIL.COM - 2006

descargarlo si ya lo estuviera. La intención es evitar tr6fico de red innecesario. Vamos a examinar el código que se encarga de esta ~rea.

Los tiempos de archivo son una de las razones de la utilización de funciones de PTP en lugar de una simple ~!_ocaciónna un,. ~nci~ de archivo. Las funciones de archivo pueden leer y, en algunos casos, escribir archivos sobre interfaces de red pero la mayorfa de lu funciones de estado, como filemtime (), no funcionan de forma remota, comportamiento que cambiad en el futuro: Esd previsto que las funciones de estado del sistema sean compatibles con ftp: *11* en PHPS'.1. De está forma, podriamos volver a implementar esta secuencia de comandos por medio de una sencilla construcci6ncopy($remoteurl, $localurl).

En primer lugar, comprobamos si tenemos una copia local del archiYo con ayuda de la función file_exiata (>.En caso negativo, tendríamos que proceder a descargar el archivo. Si existiese, obtenemos la dltima fecha de modificad6n utilizando la fund6n filemtime () y la almacenamos en la variable $local time. Si no existiese, est.blecem~ la variable $local time en o para que resulte más antigua que cualquier hora de modificación de archivos remotos:

```
ecbo 'Cbecking file ti..... cbr />';
}f 1file_exieu ($localfile))

   $localti.. • filemti .. ($localfile1l
   ecbo 'Local file laet updated •1
   ecbo datel•G,i  :l·M·Y•,  $localtiM);
   ecbo •cbr />' 1
}
•]••
   $localti..•0;
```

(Puede leer más acen:a de las funciones file_exiats O y filemtime O en varios capítulos anteriores.)

Tru establecer la hora local, necesitamos obtener la hora de modificación del archivo remoto. Para ello podemos utilizar la fund6n ftp_mdtm(>:

```
$re110teti .. • ftp_.Stal$conn, $reaotefile)1
```

Esta función toma dos par6metros (el identificador de la conexión PTP y la ruta al archivo remoto) y devuelve la marca de tiempo de UNIX de la hora a la que se modificó por dltima vez el archivo o ·1 si se produjo un error de algdn tipo. No todos los servidores FTP admiten esta fund6n, por lo que es posible que no obtengamos un resultado dtil de esta función. En este caso, hemos optado por establecer arhficialmente la variable $remotetimepara que sea más reciente que la variable ~local time agregútdole una unidad. De esta forma, nos aseguramos de que se antental'lldescargar el archivo una vez:

```
if (!$remoteti .. >• 0I)
{
   // •o significa que el archivo no est6 abl, eino que el servidor puede
   // no admitir la funci6n
   ecbo •can•t accees reaote file tiM.cbr />'l
```



```
   $reaotetiM•$localtiM+1;/t ae-.,0r••• de que exiete una actualisaci6n

ebe
{
   ·ec:bo 'Reaote file laet updated ',
   ecbo date( ..,i :1-N·Y', $:reaoteu..,,
   ecbo ·cbr />';
```

Cuando tenemos ambas horas, las comparamos para comprobar si necesitamos descargar el archivo:

```
if III$reaotetiM > $localti..)I
{
   ecbo 'Local copr ie up to date.cbr />'1
   u:it;
```

Descargar el archivo

En esta fase intentamos descargar el archivo desde el servidor:

```
ecbo 'Getting file froa eerver•'' cbr />';
$fp • fopen 1$localfile, •v•)1
if 11$eucc••• • ftp_fget1$conn, $fp, $reaotefile, nP_8IIIAIIY) )i,.
{
   ecbo 'lrror: C:0uld not dovnload file';
   fcloHl$fp);
   ftp_quitl$conn);
   e:dtl
)
fcloeel$fpl1
ecbo 'File dovnloaded eucceeefully';
```

Abrimos un archivo local utilizando fopen () como hicimos anteriormente. Tras ello, llamamos a la fund6n f.tp_fget (> e intentamos de&CJrgar el archivo y almacenarlo en una ubicad6n local. Esta función toma cuatro parúnetros. Los tres primeros son daros: la conexión FTP, el identificador de archivo local y la ruta huta el archivo remoto. El cuarto es el modo FTP. Existen dos modos de realizar una transferencia FTP: ASCil y binario. El modo ASCil se utiliza para transferir archivos de texto (es decir, archivos que se componen dnicamente de caracteres ASCII) y el modo binario, se utiliza para transferir el resto. El modo binario transfiere los archivos sin modificar, miéntru que en el modo ASCil se traducen los retornos del carro y los saltos de línea en los correspondientes caracteres para su sistema (\nen Unix, \r\nen Windows y \r en Macintosh).

La biblioteca~ de PHP incluye dos constantes predefinida, FTP_ASCII y FTP_BINARY que representan estos dos modos. Necesita decidir qué modos encajan en su tipo de archivo y puar la constante correspondiente a ftp_fget () como cuarto par6metro. En este caso vamos a transferir un archivo zip, por lo que hemos utilizado el modo FTP_BINARY La fundón ftp_fget () devuelve truesi todová bien o false si aparece un error. Almac:enamos el resultado en $aucceH e indica:

EL_HOLISTICO@HOTMAIL.COM - 2006

remos al usuario cómo fue. Tras intentar la descarga, cerramos el archivo local utilizando la función fclose (l. Como alternativa a ftp_fget (l, podríamos haber utilizado ftp_get (), que tiene la siguiente sintaxis:

```
iat ftp_get (iat coaez1da_tcp. •tring rvt•_•rch1vo_lacal,
   •tria9 rvi:•_•rc:b1vo_re_to, iat modal
```

Esta función funciona prácticamente de la misma forma que ftp_fget (l, con la diferencia de que no requiere que se abra el archivo local. Se le pasa el nombre de archivo del sistema conespondiente al archivo local que le gustaría escribir en lugar de un identificador a un archivo.

Tenga en cuenta que no hay equivalente al comando mget de FI'P, el cual se puede utilizar para descargar varios archivos a la vez. En su lugar debe utilizar varias llamada a ftp_fgec() o ftp_get().

Cerrar la conexión

Tras terminar con la conexión Fl'P, deberla cerrarla utilizando la función ftp_quit (l:

```
ftp_quit (S-J  /
```

Debería pasar a esta función el identificador correspondiente a la conexión FI'P.

Cargar archivos

Si desea realizar la operación inversa, es decir, copiar archivos desde nuestro servidor a un equipo remoto, puede utilizar dos fundones que resultan b'5icamente opuestas a ftp_fget () y ftp_get (). Estas funciones se denominan ftp_fput (l y ftp_put (l, y su sintaxis es la siguiente:

```
int ftp_fput (iat coaezl~~_!tp, striag rvca_archlvo_r-.oco.  int fp, iat II0dol

int ftp_put (int coae.1d~_tcp. •triag rui:•_•rdllvo_r•moto,
     .striag rvt•_•rcb1n,_lOC9l, int modal
```

Los parimetros son los mismos que los equivalentes de _get.

Evitar tiempos de espera

Uno de los problemas que pueden surgir al utilizar FTP para servir archivos es superar el tiempo miximodeejecución. Sabñ cumdo ocurre porque PHP deYUelve un menNjede error. Este problema suele producirse si su servidor se ejecuta en una red lenta o con mucho tráfico o si se es~ descargando un archivo de gran tamaño, como un vídeo musicaL E valor predeterminado uignado al tiempo de ejecución múúno pua todas las secaenc:ias de comandos PHP ae define en el archivophp. ini. De manera predeterminada es de 30 segundos. El objetivo es capturar secuencias de

comandos que se ejecuten fuera de controL Sin embargo, al servir archivos con Fl'P, la transferencia de archivos podria superar este intervalo de tiempo, si su cone.xión es lenta o si el archivo es grande. Afortunadamente, podemos modificar el tiempo máximo de ejecución para una secuencia de comandos dada con la función set_time_limit (). La llamada a esta funci6n restablece el número máximo de segundos que se puede ejecutar la secuencia de comandos, a partir del momento en el que se llamó a la función. Por ejemplo, si llama a

```
••t_ti.._liait(90);
```

la secuencia de comandos podri ejecutane otros 90 segundos a partir del momento en el que se llamó a la función.

Utilizar otras funciones de FTP

PHP incorpora otras funciones útiles de FI'P. La función ftp_aize () puede indicarle el tamai\o de un archivo en el servidor remoto. Su sintaxis es la siguien~:

```
int ftp_•i&e(int  coaex1da_tcp, •tring rvca_arch1vo_r1N10tol
```

Esta función devuelve el tamaAo del archivo remoto en bytes o -1 si surge un error. No todos los servidores FTP admiten esta función.

Esta función resulta útilpara determinar el tamaño de ejecución máximo que establecer para una transferencia dada. Dado el tamaño del ardiivó y la velocidad de su conexión, se puede determinar aproximadamente el tiempo que tardaré la transferencia en realizarse para utilizar la función Ht_time_limit () en consecuencia. Puede obtener y mostrar una lista de-archivos de un directorio en un servidor Fl'P remoto utilizando el siguiente código:

```
Slining  • ftp_nlin  ($COM,  "$dinccory_patll•J;
fonac:h 1$11•tlag  •• $fileam111J
   echo "$filename  cbr /~•;
```

Este código utiliza la función ftp _nliat (l para obtener una lista de los nombres de archivo de un directorio concreto. En términos de otras funciones FTP, pricticamente todo lo que hagamos desde una linea de comandos Fl'P, se puede hacer utilizando funciones Fl'P. Si desea conocer las funciones específicas de cada comando Fl'P, consulte el manual enlfnea dePHPenhttp: //php.net/manual/en/ref. ftp.php. La excepción es mget (varios get), pero puede utilizar ftp_nliat (l para obtener una lista de archivos y, a con~uación, tomar los deseados.

Lecturas adicionales

En este capítulo heinos cubierto muchos temas y, como cabria esperar, se han dejado muchos aspectos en el tintero. Si desea obtener mú informaciónsobre pro-

EL_HOLISTICO@HOTMAIL.COM - 2006

tocolos concretos y su funcionamiento, puede consultar sus RFC en ht tp: / /www. rfc-edi tor. org.

Puede que tambit!n encuentre interesante la información sobre protocolos recogida por el"consorcio de la WorldWide Webl!I\ http://www. w3.org/Protocola.

Tambit!n puede ccmsultar una obra sobre TCP/IP como *ComJ1Utn Netr«nts* escrita por Andrew·Tanenbaum.

A continuación

En el siguiente capítulo examinaremos las bibliotecas de PHP relacionadas con funciones de fecha y calelldario. Veremos cómo convertir formatos de PHP introducidos por el usuario a formatos de MySQL y viceversa.

EL_HOLISTICO@HOTMAIL.COM - 2006

20
Administrar la fecha y la hora

En este capitulo, veremos cómo comprobar y aplicar fonnato a la fecha y a la hora, y cómo pasar de un formato de fecha a otro. Esta operación resulta especialmente importante al convertir formatos de fechas entre MySQLy PHP, entre Unix y PHP, y en fechas introducidas por el usuario a través de un formulario HTML.

En este capítulo; examinaremos los siguientes aspectos:

- Obtener la fecha y la hora en PHP
- Convertir entre formatos de fecha de PHP y MySQL
- Calcular fechas
- Utilizar funciones de calendario

Obtener la fecha y la hora en PHP

En un capitulo anterior hablamos sobre el uso de la función date () para obtener y aplicar formato a la fecha y a la hora desde PHP. En este capitulo abordaremos estos aspectos y algunas otras funciones de fecha y hora de PHP.

Utilizar la función date()

Como es posible que recuerde, la función date (l toma dos parámetros, uno de ellos opcional. El primero es una cadena de formato y el segundo, de carácter

EL_HOLISTICO@HOTMAIL.COM - 2006

opcional, es una mara de tiempo de UNIX. Si no se especifica una mara de tiempo, la función date () tomará la fecha Y la hora actual de manera predeterminada. Esta función devuelve una cadena con formato en la que se representa la fecha pertinente.

Una llamada tfpica a la función de fecha podría ser:

```
ecbo datel'jS F Y');
```

El resultado de esta función será 29th Auguat 2000. En la tabla 20.1 se recogen los códigos de formato aceptados por la fecha.

Tabla 20.1. C6dlgos de formato para la función date() de PHP.

a	Maftana o tarde, representado por dos caracteres en mln.:laculaa. - o pm.
A	Maftana o tarde, representado por dos caracteres en may.:laculas, AM o PM.
B	Hora de Internet de Swatch, un sistema horario universal. SI desea més Informacl{)n al respecto, dlrfjase a http://www.swatch.com.
e	Fecha ISO 8801. La fec:ha se representa como AAAA-MM-DD. Una T mayúscula separa la fec:ha de la hora. La hora se representa en formato HH:MM:SS. Por .:lltimo, la zona horaria se representa como desplazamiento de la hora de Greenwich (GMT); por ejemplo, 2004·03-26T21:04: 42 +11: oo. (Este c6dlgo de formato es una de las novedades de PHP5.)
d	Ola del mes en formato de dos dlgltos con un cero a la Izquierda. Los valores posibles van da 01 a 31.
D	Ola de la semana en formato abreviado de dos caracteres. Los valores posibles van de Lun a Dom.
F	Mes del ano ~ formato de texto completo. Los posibles valores van de Bnero a Diciembre. ...
g	Hora del dla en formato de 12 horas sin ceros a la Izquierda. Los posibles valores van de 1 a 12.
G	Hora del dla en formato de 24 horas sin ceros a la izquierda. Los posibles valores van desde o a 23.
h	Hora del dla en formato de 12 horas con ceros a la izquierda. Los posibles valores van de 01 a 12.
H	Hora del dla en formato de 24 horas con ceros a la izquierda. Los posibles valol'N van desde o a 23.
i	Minutos del dla en formato de 24 horas con ceros a la izquierda. Los posibles valores van deecle oo a 59.
I	Horario de verano representado por un valor booleano. Devolveré 1 si la fecha se lncl~ en el horario de verano y o de lo'contrario.

j	Ola del .,.. en fonna da nOmero sin C8IOI a la Izqulenla. Los posibles valores van desde 1 a 31.
l	Ola da la semana en tDnnalD da texto completo. Los poslblea valores van desde LWI.. a Dollingo.
L	Airo bisiesto, ... par un valor booleano. Devolveré 1 11 la fecha se Incluye en un ano blsleito y o en C8IO lo contrario.
•	Mes del ano en fanna de nl'imero de dos dlgltos con ceros a la Izquierda• Los posibles valores van desde 01 a 12.
M	Mes del ano en fonnalO de 19Xto abreviado de tres caracteres. Los posibles valores vandade Bm a Dic.
n	Mes del ano en tonnato da m1mero sin caros a la Izquierda. Los posibles valores van desde 1 a 12.
o	Diferencia entre la zona horaria actual y la hora seg.:ln el meridiano de Greenwtch. Por ejemplo, +1600.
r	Fecha y hora con formato de RFC822, por ejemplo, Mi•, 9 Oct 2002 18: 45: 30 +1600. (Agregada en PHP 4.0.4.)
•	Segundos que .,._. del minuto con e.ros a la Izquierda. Los posibles valores van desde oo a 59.
S	Sufijo ordinal para fec;has con formato de dos caracteres.
t	N.:lmero total da dlaa del mes. Los valores posibles van desde 28 a 31.
T	Zona horaria del servidor en formato da tres caracteres, por ejemplo, BST.
u	N.:lmero total de segundos desde el 1 de enero de 1970 hasta este momento; tambl6n c:anoc:ldo como man:a da tiempO de UNIX.
w	Ola de la semana en fonnalD de un IOlo dlglto. ~ valores posibles van desde o (domingo) a 6 (lébado).
II	N.:lmero de semana denlrO del ano. compatible con la norma lSO-8801. (Agregada en PHP 4.1.0.)
y	Airo en formato de dos dlgltos, por ejemplo, os.
y	Airo en formato da 4 dlgltos, por ejemplo, 2005.
s	Ola del ano en formato da n.:lmero. Los valores posibles van de o a 365.
Z	Diferencia con respecto a zona horaria actual an segundos. Los valores posibles van de -n200 hasta 43200.

Marcas de tiempo de Unix

El segundo parámetro de la fundóndate () es una mara de tiempo de Unix. La mayor parte de los sistemas Unix almacenan la hora actual *y* la fecha en forma de

EL_HOLISTICO@HOTMAIL.COM - 2006

número entero de 32 bits que contiene los números de segundos desde la media noche del 1 de enero de 1910, GMT, lo que también se conoce como Unix Epoch. Si no conoce este sistema puede parecerle extrai\o al principio, pero se trata de un estmdar.

Las marcas de tiempo de Unix son una forma compacta de almacenar una fecha y una hora. Este sistema no se ha visto afectado por el efecto del ai\o 2000 como ocune con otros formatos de fecha compacta o abreviada. No obstante, presentan un problema similar, ya que sólo pueden representar un intervalo de tiempo limitado por medio de un entero de 32 bits. Si su software tiene que trabajar con eventos anteriores a 1902 o posteriores a 2038, experimentari problemas similares.

En algunos sistemas con Windows, el intervalo es más llmitado. Una marca de tiempo no puede ser negativa, por lo que las marcas de tiempo anteriores a 1910 no se pueden utilizar. Para que nuestro código sea port¸til, tendr, que tener este hecho en mente.

No tendr, que preocuparse por si utiliza su software en 2038. Las marcas de tiempo no tienen un tamaito fijo, sino que se vinculan al tamat\o de una C de gran tamai\o, que tiene al menos 32 bits. La solución mú probable es que para el año 2038, su compilador utilice un tipo de mayor tamai\o.

Aunque esté ejecutando PHP en un servidor de Windows, éste seguiri siendo el formato utilizado por la función date (l y varias otras funciones de PHP. La única diferencia es que, en Windows, la marca de tiempo debe ser positiva.

Si desea convertir la fecha y la hora a una marca de tiempo de Unix, puede utilizarla.funci6nmk.time.U, cuya sintaxis--.la 1i~

```
ini llkti-  (int Hor•.  iDt •inuto. int •eguado,   int -• ·
        inc: df•,  int do (,  int Ho.r_ver) l
```

Los par,metros resu.ltan bastante ctaros con la excepción del último, *hor_wr,* que representa si la fecha se corresponde con el horario de verano o no. Puede establecer este par'metro en 1 para seleccionar el horario de verano o 0, en caso lo contrario (el valor predeterminado es -1). Este valor es opcional y apenas se utiliza.

El problema principal que presenta esta función es el orden de los parimetros, que resulta poco intuitivo. Ademú este orden no se presta a dejar la hora sin establecer. Si no quiere establecer la hora, puede utilizar ceros en los parimetros hora, minuto y segundos. En cualquier caso, no es necesario definir los parimetros rn'9 a la derecha. Si deja los par¸metros en blanco, se les asignar6n los valores actuales. Por lo tanto, una llamada como

```
·~~~        .... ti-();
```

dewolven la marca de tiempo de Unix correspondiente a la hora y la fecha. Por ~· también puede obtener el mismo resultado llamando a

```
tttr:.·~  • t1•11;
```

La ÍlmCWII 'ti• l› no adopta ningún parimetro y siempre devuelve la marca de tiempo Unix de a. fecha y hora actuales.

Otra opción, como ya hemos mencionado, es la función date (>. La cadena de formato •u• requiere una marca de tiempo. La siguiente instrucción equivale a las dos anteriores:

```
$timeatamp • date(•U•);
```

Puede pasar un afto en formato de dos o cuatro dígitos a la función mktime (). Los valores de dos dígitos desde el 0 al 69 se interpretarán como los ai\os 2000 a 2069 y los valores del 70 al 99 se interpretar6n como los años 1910 a 1999.

Veamos otros ejemplos para ilustrar el uso de mlttime <>:

devuelve el mediodía de la fecha actual.

```
Sti . . . llkti—10.0.0.1,11,
```

devuelve el 1 de enero del año actual.

Tambien se puede utilizar mltt ime ( ) para realizar sencillas operaciones aritméticas de fecha. Por ejemplo,

```
$ti—•  llkti-112.0.0,s.on,Sday+J0,Sye•rl:
```

suma 30 días a la fecha especificada en los componentes, aunque ($day+30) ser, superior al número de días de dicho mes.

Utilizar la función getdate()

Otra función para la determinación de la hora que podría resultarle útil es la función getdate <>.Esta función consta de la siguiente sintaxis:

```
array getdate lint  • rc•_de_tiempol
```

Esta función toma una marca de tiempo como parimetro y devuelveuna matriz asociativa que representa las partes de dicha fecha y hora como se muestra en la tabla 20.2.

Tabla 20.2. p.,.. de clave y valor de la matriz aoc:illllva de la función getdate().

eeconde	Segundos, num6rlco
minute•	Minutos num•rtco
houre	Horas, num6rlco
mday	Ola del mn, num6rlc:a
wday	Ola ~ la Nffl8ll8. num6rlco

EL_HOLISTICO@HOTMAIL.COM - 2006

mon	Mes, num6rlco
year	Allo, num6rlco
yday	Ola del allo, num6rlco
weekday	Ola de le semane, formato de texto num6rlco
month	Mes, formato da texto completo
0	Marca de tiempo, num6rlco

Una vez incluidas estas partes en la matriz, se pueden procesar en cualquier formato. El elemento 0 de la matriz (la marca de tiempo) puede parecer que no sirve para nada, pero si invoca getdate (I sin parámetros, devolverá la marca de tiempo actual.

Validar fechas

Puede utilizar la función checkdate (I para comprobar si una fecha es válida. Esta función resulta especialmente útil para comprobar las fechas introducidas por el usuario. La función checkdate (I tiene la siguiente sintaxis:

```
int checkdate (int •••, int d!a, int .ao)
```

Esta función comprueba si el af\o es un número entero válido situado entre 0 y 32767, si el mes es un entero entre 1 y 12, y si el día dado existe en dicho mes. La función tiene en cuenta los af\os bisiestos.

Por ejemplo:

```
checkdate!9, 18, 19721,
```

devolverá true mientras que

```
checkdate(9, 31, 2000)
```

no lo hará.

Convertir entre formatos de fecha de PHP y ,\tySQL

~s f«h.ls Y las horas en MySQL se procesan en formato ISO 8601. Las horas funa~nan de forma rela~vamente normal, pero ISO 8601 espera a que se introduzca pnmero el al\o. Por eiemplo, el 29 de agosto de 2005, se introduciría como 2005-

08-29 o como 05-08-29. Las (echas recuperadas desde MySQL también adoptadn este orden de 111111era predeterminada.

Por lo tanto, para comunicar PHP y MySQL necesitaremos realizar alguna operación de convelSión.defechas, la ~~1 IM!p~ede realizar en cualquiera de los dos extremos.

Al introducir fechas dentro de MySQL desde PHP, puede colocarlas de forma sencilla en el fonnato correcto utilizando la función date O mostrada anteriormente. Procure utilizar las versiones de fecha y hora con ceros a la izqüierda para no confundir a MySQL. Puede utilizar un ai\o de dos dígitos pero el uso de af\os de cuatro dígitos es más recomendable. Si opta por realizar la conversión en MySQL, puede utilizar dos funciones útiles DATB_PORMAT' 1 y UIJIX_TIMBSTAMP').

La función DATB PORMAT 1) funciona de forma similar a la de PHP pero utiliza códigos de formato diferentes. En EEUU, lo habitual es aplicar el formato MM·DD-AAAA en lugar del formato ISO (AAAA-MM-DD) propio de MySQL Para ello, puede escribir la consulta de la siguiente forma:

```
SSLSCT~ft l'OIIIAT(columna ~-ba .,. Id IA')
PRCIN nooe11n:de_tabla1 -
```

El código de formato lm representa el mes en forma de número de dos dígitos; Id, representa los dfas en forma de número de dos dígitos; y U representa el af\o en forma de número de 4 dígitos. En la tabla 20.3 se recogen los aSdigos de formato MySQL más útiles.

Tabla 20.3. C6dlgo de formeto para la funcl6n OATE_FORMAT() de MySQL.

IN	Mes, texto complal0
IW	Nombre del dla da la semana, texto completo
ID	Ola del mes, num6rlco, con sufijo de texto
IY	Allo, num6rlco, 4 dlgltoa
ly	Allo, num6rlco, 2 dlgltoa
la	Ola de la semana, 3 caractents
Id	Ola del mu, num6rlco, con c:eros a la izquierda
le	Ola del mes, num6rlco, sin ceros a la lzquierda
la	MH, numérico, con ceros a la Izquierda
le	MH, num6rlco, sin ceros a la izquierda
lb	Mes, texto, 3 caraclentl
lj	Ola del allo, num6rlco
IH	Hora. reloj de 24 horas, con ceros a la lzquierda

EL_HOLISTICO@HOTMAIL.COM - 2006

tk	Hora, reloj de 24 llores. sin ceros a la izquierda
IbOU	Hora, reloj de 12 llena. c:on ceros a la Izquierda
11	Hora, reloj de 12 horas. sin ceros a la Izquierda
ti	Minutos, numslcoa, con ceros a la Izquierda
Ir	Hora, en fol'IIIIIID de 12 horas (hh:mm:ss [AMIPM))
IT	Hora, en formato de 24 horas (hh:mm:ss)
IS o ta	Segundos, num6rtco, ceros a la izquierda
tp	AMoPM
tw	Ola de la semana, num6rtco. de 0 (domingo) a 6 (úbado)

La función UHIX_TIMBSTAMP funciona de forma similar, pero convierte una columna en una marca de tiempo de Unix. Por ejemplo,

```
SBLBCT 0lfIX_TIMBSTlUIPCcolumna~tecllal
!'RON 'aombre_de_tllble:
```

devolverá la fecha con formato de marca de tiempo de Unix. En PHP puede hacercotnote parezca.

Como regla general, utilice una marca de tiempo de Unix para los cálculos de la fecha y el formato de fecha estándar al almacenar o mostrar fechas. Resulta más sencillo realizar cálculos de fechas y comparaciones con ~cas de tiempo de Unix.

Calcular fechas en PHP

La forma más sencilla de calcular el intervalo de tiempo que discurre entre dos fechas en PHP consiste en utilizar la diferencia entre marcas de tiempo de UNIX. Hemos utilizado este enfoque en la secuencia de comandos ilustrada en el listado 20.1.

UatNo 20.1. calc_age.php. Esta secuencia de comandos calcula la edad de una persona a partir de su fecha de nacimiento.

```
c7p11p·
   // ••teblezca la fecha para el c6lculo
   Sday • 11;
   '-tb • 9;
 ~$7eu "7 19'12;

   // ~•rde que aeceaita el dla de aaciaieato en foraato d~a. -• y aAo
   Sbdayuaix • aktiae 10, o, o, $moatb, Sday, $yearl;
   // obcaaga la -rea de tieapo
```



```
   snowuaix • tiae¢,,
   // obtenga la marca de tiempo de unix corre11¡10ndiente al dla de boy
   $ageunix • Snowunix - $bdayunix, // calcule la difarancia
   Saga• floorl$ageunix / 1315 • 24 • 10 • 1011
   // convierta loa segundo• en aftoa

   echo •11,ge ia $a9a•1
```

En esta secuencia de comandos, hemos establecido la fecha para el cálculo de la edad. En una aplicación real, es probable que esta información proceda de un formulario HIML En primer lugar, llamamos a la función mktime () para calcular la marca de hora correspondiente al cumpleat\os y a la fecha actual:

```
$bdayunix • mkti— (O, O, O, $aonth, $day, $year),
$ftOVWliX • mktiae()¡
// obtenga la aarca de tieapo de unix correspondiente al dla da hoy
```

Una vez que tenemos estas fechas en el mismo formato, bastará con restarlas:

```
Sageunix • $nowunix - $bdayunix:
```

Ahora viene la parte complicada: convertir este resultado en una unidad de medida que resulte más sencilla para los humanos. Podemos volver a convertir los at\os dividiendo el resultado por el número de segundos de los que consta un at\o. • Seguidamente redondeamos el resultado utilizando la función floor ‹ I ya que una persona no tendrá, por ejemplo 20 81\os, huta que los cump!,a:

```
Saga• floor(Sageunix/ (355 • 24 • 50 • 5011: // convierta loa aegundos en aftoa
```

Tenga en cuenta que este enfoque presenta el problema del Ífmite que impone el rango de las marcas de tiempo de Unix (por regla general, enteros de 32 bits). Puede que este ejemplo no sea una aplicación ideal para utilizar marcas de tiempo ya que sólo se puede utilizar para personas nacidas a partir de 1970.

Calcular fechas en MySQL

PHP no incluye, de manera predeterminada, funciones de manipulación de fechas. Evidentemente, podemos crear las nuestras propias pero puede resultar complicado tener en cuenta aspectos como-los añoe bisiestos o loe cambios horarios. Otra opción consiste en descargar las funciones de otros usuarios. En el manual de PHP encontrará las contribuciones de otroll usuarios, pero sólo algunas son recomendables.

Una posibilidad no tan evidente éonsiste en utilizar MySQL MySQL proporciona una gran variedad de funciones de manipulación de fechas para trabejar con periodos temporales que superen el ámbito de marcas de tiempo de Unix. Tendrá que conectarse a un servidor de MySQL para ejecutar una consulta MySQL, pero rio tendrá que utilizar los datos de la base de datos.

EL_HOLISTICO@HOTMAIL.COM - 2006

La siguiente consulta suma un día a la fecha 28 de febrero de 1700y devuelve la fecha resultante:

```
•elect adddatel'1700·02·28', interval 1 d9Yl
```

El al\o 1700no es bisiesto, por lo que el resultado es 1100-03-01.

Enelmanualdet.t~L,http://www.mysql.com/doc/en/Date_and_time_functions.html, encontrará una detallada sintaxis para describir y modificar fechas y horas.

Desafortunadamente, no existe una forma sencilla de obtener el número de aftos comprendido entre dos fechas, por lo que el ejemplo de la fecha de nacimiento sigue presentando ciertos fallos.

Resulta muy sencillo obtener la edad de una persona en días y el listado 20.2 convierte dicha edad en aftos.

LI•taclo 20.2. mysql_c:alc_age.php. Utilizar MySQL pera determinar la edad de una persona en función de •u fecha de nacimiento.

```
c?php
   // defina la fecha para el c&lculo
   $day • 18l
   Smonth • 91
   $year • 19721

   11 aplique fo~to a la fecha de nacimiento COIIID facha ISO 8'01
   $bdayISO • date(•c•, lllktime (0, 0, 0, $month, $day, $yearll1

   // utilice una coneulta ayeql para calcular la edad en dlaa
   $db • iar•qli_connectl'localhoat•, •u•er•, •pa••'l1
   $re•• my•qlI_query($db, •aelect datedifflnowll, '$bdayISO'l"l1
   $age • my•qli_fetch_array($re•>1

   // convierta la edad en dlae en alloa (aproxi..da... te)
   echo •11ge ia •.floorl$age[0)/3&5.25)1
```

Tras aplicar formato a la fecha de nacimiento como marca de tiempo ISO, pasamos la siguiente consulta a t.tySQL:

```
aelect datediff(now(), '1972·09·11T00:00:00+10:00')
```

La función now() de t.tySQL siempre devuelve la fecha y la hora actual. La función datedif f () de t.tySQL (incluida desde la versión U.1) resta una fecha de otra y devuelve la diferencia en días.

Como no existen funciones incorporadas específicas para realizar estos cálculos, la consulta SQL para calcular el nllmero exacto de aftos es un tanto compleja. En este c~, dividimos la edad en días por 365,25para obtener la edad en al\os.

Este cálculo puede variar por un al\o si se ejecuta en la fecha de nacimiento de alguien, en función de cuántos al\os bisiestos hayan pasado en la vida de dicha persona.

Utilizar fllkrosegundos

En algunas aplicaciones, la medición del tiempo en segundos puede que no sea lo suficientemente precisa para que resulte lltil. Si desea medir periodos muy breves, como por ejemplo lo que tarda en ejecutarse una secuencia de comandos de PHP, tendrá que utilizar la función microtime ().

Si utiliza PHPS, debe invocar microtime () con el parámetro get_as_float establecido en true. Esta invocación devuelve una marca de tiempo como número de coma flotante para que lo pueda utilizar como desee. La marca de tiempo es la misma que devuelven las funciones mktime (), time (› o date<), pero con un componente fracdonal.

La instrucción

```
echo number_formatl•icroti..ltrual, 10, '.', '');
```

genera un resultado similar a 1080303003,1321949959.

En versiones anteriores no se puede solicitar el resultado como número de coma flotante, sino que se proporciona como cadena. La invocación de microtime ‹) sin parámetros devuelve una cadena de tipo •0 , 0214 93 0 0 1 o e o 3 o~•. El primer número es la parte fracciona! y el segundo es el número de segundol transcurridos desde el 1 de enero de 1970.

Resulta mb sencillo trabajar con números que con cadenas, por lo que si no le importa que su código requiera PH 5.0 para ejecutarse, es mú sencillo invocar microt ime () con el parámetro true.

Conviene destacar que no estamos seleccionando datos de una tabla y que ni siquiera seleccionamos una base de datos para utilizarla en esta secuencia de comandos, pero no es necesario iniciar sesión en el servidor t.tySQL con un nombre de usuario y una contrasefta válidos.

Utilizar funciones de calendario

PHP consta de una serie de funciones que permiten convertir fechas entre dife.. rentes sistemas de calendario. Los calendarios principales con los que se suele trabajar son el calendario gregoriano, el calendario juliano y la cuenta de días juliana.

El calendario gregoriano es el m4s extendido actuslmenteen los países occidentales. El 15 de octubre de 1582del calendario gregoriano equivale al 5 de octubre de 1582 en el calendario juliano. Antes de dicha fecha, se solía utilizar el calendario juliano. Cada pafs pasó al calendario gregoriano en una fecha distinta y alguna&no lo hicieron hasta principios del siglo XX.

Aunque es posible que haya oído hablar de estos dos calendarios, puede que no sea así con el tercero, la cuenta de dfas juliana. Este calendario se parece mucho a la marca de tiempo de Unix. En este calendario se cuenta el número de días desde una fecha en tomo al afto 4000 antes de Cristo. En sí mismo, no resulta especialmente

EL_HOLISTICO@HOTMAIL.COM - 2006

útil. pero• puede atilizar pua realizar conffllione entre fonnatas. Para pasar de un formato a otro, prilmro se convmte la Cedla a la cuenta de clfas juliana y c:les-pu& • pasa el re.ultado al calendario deNado.

Para utilizar ..... funcioMs con Unix, neallitahaber compilado la extensi6n del calenclario en PHP. a.a.• incorporan en la ilwblwimcie Windows estúdar.

Para hacene una idea.~ las siguientes sintaxis de las fuJICiow que • utilizanpara convertir fechas del c:aJendario gregoriano al calendario juliano:

```
illt ——riaDtojd (iat —•· iat dl&, iat dol
.crillg jdtojuliaaCiDt dlajuJ.J-1
```

Para convertir una fecha, necesitaríamos llamar a ambas funciones:

```
$jd • gre,aoriaatojd (t, is, 15121;
ecba jdtojuliaal$jdl
```

P.ste c6digo devuelve la fecha juliana en formato mm/ dd/aaaa.

Exi1tm1 variaciones de eetu funciona para realizar conveniones entre el calendariopegmimo, juliano. .fnncé y judfo, a muas de tiempo de Unix.

Leduras adicionales

Sia...--mú.,.. lufuildoa.-de fecha hora de PHP M [redacted]
——las..._.reloantesde suamamaJ•en

```
bttp, 1(pllp.aet/-1/-/nf.dM:.cue.pllp
11t~.,~.a,ill¡L-,dlll:/ealaac....M11:-t1-_~1-.-.L.
```

Si va á realizar una conversión entre caJenc:luioe,c:omulte la i-gina del manual correspondiente a lu funciolles de calendario de PHP (ht tp ://pbp. net /manual/en/ref.calendar.php).

A continuación

Una de lu funciones má útiles de PHPes su capacic:lad para crear ilúgenes al instante. En el siguiente capítulo• camentar6 CC5mo utilizar la funci6n de biblioteca de imapn·para lograr algunoe efectos útiles eintmesmtta

EL_HOLISTICO@HOTMAIL.COM - 2006

Una de las funciones más útiles de PHP es su capacidad para crear imágenes al instante. PHP incorpora varias funciones de información sobre imágenes y permite utilizar la biblioteca GD2 para crear nuevas imágenes o manipular la existentes. En este capítulo veremos cómo utilizar las funciones de imagen para conseguir efectos útiles e interesantes. En concreto examinaremos los siguientes aspectos:

- Configurar la compatibilidad para imágenes en PHP
- Comprender los formatos de imagen
- Crear imágenes
- Utilizar imágenes generadas automáticamente en otras páginas
- Utilizar texto y fuentes para crear imágenes
- Dibujar figuras y representación gráfica de datos

Vamos a utilizar dos ejemplos en los que generaremos botones de un sitio Web al instante y dibujaremos un diagrama de barras utilizando datos procedentes de una base de datos de MySQL.

Utilizaremos la biblioteca G2 aunque existen otras bibliotecas de imágenes de PHP. La biblioteca ImageMagick no forma parte de la instalación estándar de PHP pero resulta muy sencillo instalarla desde PECL (Biblioteca de clases de extensiones de PHP). JmageMagick y GD2 cuentan con características similares pero en algunos aspectos resulta más indicada ImageMagick. Si desde crear imágenes GIF

EL_HOLISTICO@HOTMAIL.COM - 2006

(incluso GIF animados), le recomendamos ImageMagick. Si desea trabajar con imágenes de color verdadero o con efectos de transparencia, pruebe a comparar lo que ofrecen ambas bibliotecas. Puede consultar la PCEL correspondiente a la descarga de ImageMagick para PHP en http: / /pecl. php. net /package/ imagick.

En el sitio principal de lmageMagick encontrará demostraciones de sus prestaciones y una detallada documentación (http://www.imagemagick.org).

Configurar la compatibilidad de imágenes en PHP

Algunas de las funciones de imagen de PHP siempre están disponibles pero muchas de ellas necesitan la biblioteca GD2 que puede encontrar en ht tp: / /www. boutell. com/gd.

Desde la versión 4.3, PHP incorpora su propia versión de la biblioteca GD2. Esta ,·ersión consta de funciones adicionales y suele estar más actualizada, por lo que es aconsejable utilizarla. En Windows, las imágenes PNG y JPEG disponen de compatibilidad inherente. Si utiliza Unix y quiere trabajar con imágenes en formato PNG, deberá instalar libpng y zlib desde los siguientes lugares (respectivamente):

```
:ttp://,,...,.libpng.org/pub/png/
http~//www.gzip.org/zlib/
```

Necesitará configurar PHP con las siguientes opciones:

```
--witb-png-dir•/ruta/a/libpng
--witb-zlib-dir•/ruca/a/zlib
```

Si utiliza Unix y quiere trabajar con imágenes en fo~~to JPEG, ~ecesi.tarád~ cargar jpeg-6b y volver a compilar GD con la compatibilidad JPEG incluida. Diríjase al siguiente sitio para realizar la descarga:

```
~tp://ftp.uu.net/graphies/jpeg/
```

Seguidamente, necesitará volver a configurar PHP con la opción

```
··vitb-jpeg·dir•/path/to/jpeg-6b
```

y volver a compilarlo. Si quiere utilizar fuentes TrueType en sus imágenes, necesitará utilizar la biblioteca FreeType. Esta biblioteca se incluye en PHP 4. Puede descargarla desde ht tp: / /www. freetype. org.

Si desea utilizar fuentes PostSaipt de tipo 1, deberá descargar la b1bhoteca tllib disponible en ftp: / / sunsi te. une. edu/pub/Linux/ libs/graphics.

Tras ello necesitará ejecutar el programa de configuración de PHP con

```
--wic!l-tllibl•ruca/a/cllib)
```

Por último, tendrá que configurar PHP por medio de - -with-gd.

Formatos de imagen

La biblioteca GD admite los formatos JPEG, PNG y WBMP. Ha dejado de admitir el formato GIF. A continuación, repasaremos rápidamente cada uno de estos formatos.

JPEG

JPEG equivale a Joint Photographic Experts Group (Grupo conjunto de expertos en fotografía) y es el nombre de un conjunto de estándares. El formato de an:hivo al que se hace referencia al hablar de imágenes JPEG se llama JFIF y se corresponde con uno de los estándares del grupo mencionado.

Las imágenes JPEG se suelen utilizar para almacenar fotografías u otras imágenes con muchos colores o gradaciones de color. Este formato utiliza compresión con pérdida (es decir, que para reducir una fotografía en un archivo de menor tamaño se pierda parte de la calidad de la imagen). Como las imágenes JPEG deben conservar lo esencial de sus referentes analógicos, con las gradaciones de color, ~ ojo humano puede tolerar alguna pérdida de calidad. Este formato no resulta adecuado para dibujos lineales, texto o bloques sólidos de color. Si lo desea puede leer más sobre JPEG/JFIF en el sitio oficial de JPEG, http://www. jpeg. org•

PNG

PNG equivale a Portable Network Graphics (Gráficos portátiles de red). Este formato de archivo es el sustituto del formato GIF (del inglés Graphics Jnterchange Format, Formato de intercambio de gráficos) por razones que comentaremo1i"següidamente. El formato PNG incorpora compresión sin pérdida, por lo que resulta adecuado para imágenes que contengan texto, líneas rectas y bloques sencillos de color como encabezados y botones de sitios Web (los mismos propósitos a los que servía antes el formato GIF). Una versión comprimida de PBG de la misma imagen suele tener un tamaño similar a la versión comprimida en GIF. El formato PNG ofrece mejor compresión que el formato GIF así como transparencia variable, corrección gamma y entrelazado bidimensional. Sin embargo, no admite animaciones. Para ello, debe utilizar el formato extendido MNG, que está actualmente en fase de desarrollo. Los sistemas de compresión sin pérdida son muy indicados para las ilustraciones pero no para almacenar fotografías de gran tamaño ya que suelen producir archivos de gran tamaño. Puede leer más sobre el formato PNG en su sitio oficial, http://www. libpng. org/pub/png.

WBMP

WBMP equivale a Wireless Bitmap (Mapa de bits inalámbrico). Se trata de un formato de archivo diseñado específicamente para dispositivos inalámbricos.

EL_HOLISTICO@HOTMAIL.COM - 2006

GIF

GIF equivale a Formato de intercambio de gráficos. Se trata de un formato comprimido sin pérdida utilizado en la Web paiá almacenar imágenes que contengan texto, líneas rectas y bloques de un solo color. Seguro que se está preguntado por qué GD no admite este formato. La respuesta es que solía hacerlo, hasta la venión 1.3. Si desea instalar y utilizar funciones GIF en lugar de funciones PNG, puede descargar la versión 1.3 de GD de ht tp: / /www. linuxguruz. org/ downloade/ gdl. 3. tar .gz.

Tenga en cuenta que los creadores de GD no aconsejan utilizar esta versión y han dejado de incorporar compatibilidad para el formato GIF. Esta copia de la venión GIF es probable que deje de estar disponible en el futuro.

Existe una segunda razón para no admitir imágenes GIF. El formato GIF estándar utiliza una forma de compresión conocida como LZW (Lempel Ziv Welch) que está sujeta a una patente propiedad de UNISYS. Los proveedores de los programas que leen y escriben formatos GIF deben pagar por el uso de la licencia a UNISYS. Por ejemplo, Adobe paga una cuota de licencia por los productos como Ph4?toshopque se utilizan para crear imágenes GIF. Los creadores de bibliotecas de código también deben pagar una cuota, asf como los usuarios de dichas bibliotecas. Por lo tanto, si utiliza la versión GIF de la biblioteca GD en su sitio Web, deberá abonar las elevadas cuotas que aplica UNISYS. Esta situación resulta poco afortunada porque el formato GIF llevaba mucho tiempo utilizándose antes de que UNISYS decidiera cobrar por su uso. De hecho, se convirtió en uno de los estándares de la Web. El paso dado por UNISYS no ha sentado nada bien en la comunidad de desarrollo Web. Si lo desea, puede leer más al respecto (y hacerse su propia opinión) en el sitio de UNISYShttp://www. unisys .com/about_unieye/lzw yenelsitioopuesto, BurnAllGils, en http://burnallgifs.org.

La venión norteamericana de la patente LZW caducó el 21 de junio de 2003, y la europea y la asiática durante 2004. No somos abogados y nada de lo que digamos aquf debe interpretarse como un consejo legal, pero en nuestra opinión resulta mú sencillo utilizar el formato PNG, independientemente de las distintas posiciones. En años anteriores la compatibilidad de los navegadores con PNC era variable pero ha mejorado en las nuevas versiones. La compatibilidad completa con las funciones avanzadas de PNG sigue siendo imperfecta pero válida para imágenes sencillas.

Se espera que la compatibilidad con GIF vuelva a implementarse en PHP 5.0.1 y PHP 4.3.8 (que aparecerán tras la fecha de caducidad de la patente LZW).

Crear imágenes

Los cuatro pasos básicos para crear una imagen en PHP son los siguientes:

1. Crear un lienzo sobre el que trabajar.
2. Dibujar formas e imprimir texto en dicho lienzo.

3. Generar el gráfico final.
4. La"berar los recunos.

Comenzaremos por examinar una secuencia de comandos muy sencilla para la creación de imágenes. En el listado 2U se recoge esta secuencia de comandos.

Lle~ 21.1. slmplegraph.php. ~ un gr6fico c:ompuesto de una 1enclla llnea con la secuencia de texto Sales.

```
. c?php
 // cree la iMgea
    Sheigbt • 200,
    Swidtb • 200,
    Si••  IMge<:reataTnaeColor(Swidth, $height);
    Swhita • llN19eClolorAllocata (Si•, 255, 255, 255);
    Sblack • IIN19eClolorAllocata (Si•, 0, 0, &t);

 // dibuja la lugaa
    ImagaF1111$1•, 0, 0, SbluaJ,
    IM9aLiaa($im, 0, 0, Swidth, $beight, $Wbital,
    IM9e~tril19(Si•, t, 50, 150, 'lalaa•, $wbitaJ)

 11 9eaara J:a iMgaa
    Raadar ('Coataat-typa:  iuge/pag•J,
    Ima9ePag <Si•) ;

 // libara racur90a
    I • 9•Daatroy(Si•>,
```

El resultado de ejecutar esta secuencia de comandos se ilustra en la figura 21.1.

Figura 21.1. La secuenc:la de comandos dibuja un fondo de color azul y agrega una linea y una aecuencllll de taJdo a la lmagen.

Vamos a desgranar los pasos seguidos para crear esta imagen.

EL_HOLISTICO@HOTMAIL.COM - 2006

Crear un lienzo

Para empezar a construir y cambiar una imagen en PHP, necesitará aear un identificador de imagen. Existen dos formas básicas de hacerlo. Una consiste en crear un lienzo en blanco, para lo cual se puede llamar a la función ImageCreateTrueColor (). como hemos hecho en la secuencia de comandos de la siguiente forma:

```
Si•• 1.. geCreateT~Color(Svidtb, Sbeigbtl;
```

Es necesario pasar dos parámetros a ImageCreaceTrueColor (). El primero es la anchura de la imagen y el segundo su altura. La función devot,·erá un identificador para la nueva imagen. (Su uso resulta parecido a los identificadores de archivo.)

También se puede leer un archivo de imagen existente y, a continuación, filtrarlo, modificar su tamaño o agregarlo. Para ello puede utilizar las funciones ImageCreateFromPIIG(),ImageCreateFromJPEG()oimageCreateFromGIF(), según el formato de archivo que esté leyendo. Estas funciones toman un nombre de archivo como parámetro, por ejemplo,

```
Si•• IaageCreatePrcaPRG('baaeimage.png•);
```

: En una sección posterior del capítulo se incluye un ejemplo en el que se utilizan imágenes para crearbotones alinstante.

Dibujar o imprimir texto en la imagen

La operación de dibujar o escribir texto en la imagen consta de dos fases. En primer lugar, debe seleccionar los colores que se utilizarán en el dibujo. Como es probable que ya sepa, los colores que se muestran en un monitor se componen de diferentes cantidades de luz roja, verde y azul. Los formatos de imagen utilizan una paleta de colores que se compone de un subconjunto dado de todas las combinaciones posibles de estos colores. Para utilizar un color para dibujar una imagen, es necesario agregarlo a la paleta de la imagen. Este paso se debe realizar para todos los colores que se desee utilizar, incluidos el negro y el blanco.

Puede seleccionar los colores para la imagen llamando a la función I-geColorAllocace (). Necesita pasar su identificador de imagen y los valores rojo, verde y azul del color que desee dibujar dentro de la función.

En el listado 21.1, se utilizan dos colores: el negro y el blanco. Los asignamos llamancloa

```
1.-..Ste.~  lllA98ColorAJ.locate ISim, 2s5.  255, 2551;
 ~          l8a9eColo~locate   ($1a, 0, 0, 64);
```

La lunci&a dnuel,,e un identificador de color que podemos utilizar para acceder al color poi1 1Eiiut1nente.

En segundo lugar, para dibujar la imagen. existen varias funciones disponibles, según lo que queramos dibujar: lineas, arcos, polígonos o texto.

Las funciones de dibujo requieren los siguientes parámetros:

- El identificador de imagen
- Las coordenadas de inicio y a veces las del final de lo que queremos dibujar
- El color en el que desea dibujar
- La información de fuente para el texto

En este caso, hemos utilizado dos funciones de dibujo. Vamos a examinar cada una de ellas por separado.

En primer lugar, pintamos un fondo negro sobre el que dibujar utilizando la función ImageFill ():

```
ImagePill(Sim, o. o, $blue);
```

Esta función toma un identificador de imagen, las coordenadas iniciales del área que pintar (x e y) y el color de relleno como parámetros.

Observe-que-las-eoerdenadaede-18imagen ea establecen a partir de la esquina superior izquierda, qua asx-o a y• O. La esquina inferiorderecha da la imagen asx-$width. Y•$heighc. Tanga cuidado porque esta sistema es al opuesto al qua se suele utilizar en los gráficos.

A continuación, hemos dibujado una linea que va de la esquina superior izquierda (o, 0) hasta la esquina inferir derecha ($width, $height) de la imagen:

```
ImageLine(Sim, 0, 0, Svidtb, Sbeigbt, $vbitel;
```

Esta función toma el identificador de imagen, el punto de partida de x e y correspondiente a la linea, el punto final y el color como parámetros.

Por último, agregamos una leyenda al gráfico:

```
ImageString(Sim. 4, so, 150, 'Salea•, $vbitel
```

La función ImageString () toma varios parámetros ligeramente diferentes. Su sintaxis es-la siguiente:

```
int iaageatring (reaource im, int tuente, int x, int y, atring •· int col)
```

Toma como parámetros el identificador de la imagen, la fuente, las coordena'~ x e y en las que empezar a escribir el texto, el texto que escribir y el color.

El valor de la fuente puede estabJecerse de 1 a 5, que representan un conjunto d:e fuentes incorporadas. Como alternativa, puede utilizar fuentes Tl'J.leType y

EL_HOLISTICO@HOTMAIL.COM - 2006

PoslSc:riptType 1. Cada uno de estos ccquntos de fuentes consta de sus funciones correspondientes. En el siguiente ejemplo utilizaremos las funciones TrueType.

Una buena razón para utilizar uno de los conjuntos alternativos de funciones de fuentes es que el tato escrito por la funcióll I—g•String (› asf como por las funciones asociadas, como ImageChar I) (escribe un car6c:teren la imagen), est4 sin suavizar. Las funciones de TrueType y PostSaipt producen texto suavizado.

Si no est, iegwo de las diferencias, examine la figura 21.2. En las zonas en las que aparecen curvas o líneas en mgulo el texto sin suavizar parece dentado. La curva y el mgulo se consiguen utilizando un efecto de •escalera". En la imagen suavizada, en las zonas de curvas o mgulos, se utilizan pfxeles con colores que varían entre el utilizado en el texto y el del fondo con el objeto de suavizar la apariencia del texto.

Normal

Anti-aliased

Flg... 21.2. El texto normal pwece dentado. especialmente cuando el tamafto de la fuent11 es grande. En el texto suavizado, las curva, y l81 aqutnas de las letras pierden el cartcter dentado.

Generar el gráfico final

Puede dirigir la imagen final directamente al navegador para su representación o a un archivo. En este ejemplo, vamos a representar la imagen en el navegador. Esta operación consta de dos pasos. En primer lugu, necesitamos indicarle al navegador Web que vamos a representar una imagen en lugu de texto o código H1ML. Para ello utilizamos la función Header () en la que especificamos el tipo MIME de la imagen:

```
Headllr ('Content-type, 1..ge/png'l;
```

Por regla general, al recuperu un an:hivo en el navegador, el 'tipo MIME es lo primero que envía el servidor Web. Pua una p,gina HTML o PHP, lo primero que se envía ser,

```
Content-type, text/htal
```

Esto indica al navegador cómo interpretar los datos que siguen. En este caso, queremos decirle al navegador que vamos a enviu una imagen en lugu de la salida HTML habitual. Para ello, podemos utilizar la función Header (), que todavía no hemos visto. Esta función envía cadenas de encabezado HTML sin procesar. También se suele utilizar pua realizar redireccionamientos HTTP. En esta operación, se indica al navegador que cargue una P'gina diferente en lugar de la seleccionada. Este recurso se suele utilizar al mover una Jt'gina. Por ejemplo:

```
Headllr (•Location, http,//www.dollain.c011/new_bolle_p119e.htlll'l
```


Al utilizar la función Header () tenga en cuenta que no 1e puede ejecutar si ya se ha enviado un encabezado H1TP para la P'gina. PHP enviad un encabezado HTTP autom,ticamente en cuanto se devuelva algo al navegador. Por lo tanto, si ha devuelto alguna instruc:dón echo o incluso algún espacio en blanco antes de la etiqueta PHP de apertura. se enviamn los encabezados y PHP devolved un mensaje de advertencia al intentar llamu a Header (). Sin embargo, puede enviar varios encabezados H1TP con vufas llamadas a la función Header () en la misma secuencia de comandos, aunque todas ellas deben aparecer antes de que se envfe cualquier salida al navegador.

Tras enviar los datos del encabezado, devolvemos los datoe de la imagen con una llamada a

```
1..gel'ng C$i•I;
```

Esta llamada envía el resultado al navegador en formato PNG. Si desea utilizar un formato diferente, puede llamar a IugeJPBG () (si est4 activada la compatibilidad para JPEG). También tendr, que enviar el encabezado correspondiente en primer lugar; es decir:

```
Headllr C•content-type, 1—9•/:lpe9' l l                    !$.
```

La segunda opción que puede utilizar, como alternativa a las anteiTores, es escribir la imagen en un an:hivo en lugar de hacerlo en el navegador. Para ello, puede agregar un segundo pammetro opcional a ImagePHG (o una fundón similar para el resto de los formatos compatibles):

```
I..gePIIOC$im, $filana..ll
```

Recuerde que se aplican todas las reglas habituales sobre la escritura en UI' archivo desde PHP (por ejemplo, haber configurado correctamente los permisos).

Liberar recursos

Cuando haya terminado con una imagen, debe devolver los recursos que se han estado utilizando al servidor destruyendo su identificador. Para ello puede realizar una llamada a IugeDestroy ():

```
1..g•D••troy($ial;
```

Utilizar imágenes generadas automáticamente en otras páginas

Como un encabezado sólo se puede enviar una vez y ésta es la única forma de indicar al navegador que estamos enviando datos de imagen, resulta un tanto com-

EL_HOLISTICO@HOTMAIL.COM - 2006

plicado incrustar imágenes creadas al instante en una página normal. A continuación, se indican las tres posibles opciones:

1. Puede utilizar una página completa en forma de imagen, como hicimos en el ejemplo anterior.
2. Puede copiar la imagen en un archivo como se indicó anteriormente y utilizar la etiqueta cimg> para hacer referencia a él.
3. Puede incluir la secuencia de comandos de generación de imágenes en una etiqueta de imagen.

Ya hemos visto los primeros dos métodos. A continuación, repasaremos brevemente el tercero. Para utilizar esta técnica, se incluye la imagen dentro del código HTML mediante una etiqueta de imagen situada de la siguiente forma:

```
cia,g arc•••implegraph.php• heigbt ••zoo• widta-••zoo• alt••sale• going dolftl• />
```

En lugar de colocar una imagen PNG, JPEG o GlF, insertamos la secuencia de comandos PHP que genera la imagen en la etiqueta SRC. Ésta se recuperará y el resultado se agregará en lfnea, como se ilustra en la figura 21.3.

Figura 21.3. La imagen en linea generada dinámicamente parece igual que un11 imagen convencional para un usuario final.

Utilizar texto y fuentes para crear imágenes

Vamos a examinar un ejemplo más complejo. Resulta dtil poder crear botones u otras imágenes para un sitio Web de manera dinámica. Puede crear botones sencillos utilizando un rectángulo de color de fondo mediante las técnicas explicadas

anteriormente. Mediante programación también puede generar efectos complicados aunque puede recurrir a un programa de dibujo para ello. De esta forma el artista se puede dedicar a la parte artística y los programadores a la programación.

En este ejemplo, vamos a generar botones utilizando una plantilla de botón vaáa con efectos como bordes con relieve, que resultan mucho más sencillos de generar con Photoshop, GIMP u otras herramientas gdficas. Con la biblioteca de imágenes de PHP, podemos partir de una imagen básica y dibujar sobre ella.

Utilizar~os fuentes TrueType para poder recurrir a la función de texto suavizado. Las funciones de fuente TrueType tienen sus propios elementos especiales, que se comentarán posteriormente.

El proceso b~sico consiste en tomar una secuencia de texto y generar un botón colocando el texto encima. El texto se centrará horizontal y verticalmente en el botón, y se utilizará el tamaño más grande que resulte posible encajar en el botón.

Crearemos una interfaz para el generador de botones con fines de prueba y experimentación. En la figura 21.4 se ilustra esta interfaz. (No hemos incluido el código HMTL correspondiente a este formulario porque resulta muy sencillo, pero podrá encontrarlo en el CD en design_button. html.)

Figura 21.4. La interfaz permite a un usuario seleccionar el color del botón y eec:ribir el textodeseado.

Puede utilizar este tipo de interfaz para un programa que genere sitios Web automáticamente. También puede llamar a la secuencia de comandos escrita de manera interna y generar todos los botones de un sitio Web al instante.

En la figura 21.5 se muestra el resultado típico de esta secuencia de comandos. ~' botón es generado por une secuencia de comandos llamada -ke_button.phpf Esta secuencia de comandos se ilustra en el listado 21.2.

EL_HOLISTICO@HOTMAIL.COM - 2006

Plgunl 21.1. un bol6n gener.so.por la NCUenCIII de comandoa rmu_bullan.pllp.

Uetado 21.2. IMke_button.php. Em aecuencla ele comandos • puede IIIIIMr desde el fonnulMo clnign_bullan.html o clelde dentro de UI'III eliqum ele lmegen HTML

```
c7pbp
// compruebe qua tea•-• loe dat- de variable correcto•
// la• variabl•• aoa buttoa•text y color

Sbuttoa_text • S_IISQUIISTl'button_tut•J;
$color• $_1SQUS8T['color•J1

if CC•1111ty($buttoa_textl ll et1pty($colorll ll ![($color• •'red' ll $color•
• ':lllua' H .$calar• • 'gree11 •111
(
  ecbo •could aot creat• 1uge • far. aot Ulled out correctly•,
  ex1t;


// era• una 1..gen coa el fondo correcto y compruebe el ta•fto
$1• • i..gecreatefroiapag (Scolor.' ·buttoa.png•)1
U CISia1
(
  ecbo 'could aot craate 1..9•'1
  a1t1


' S•1dtb_1uge • Iuge&Z 1$1•> 1
Sba1gbt_1mage • IugeSYCSi•I;

// IIUHtra• 1.&g•ae• aecedtaa un urgen de 18 pixel•• de•de el borda de la iugea
Sw1dtb_1uge_wo_urgln• • $w1dtb_image • 12 • 1111
Sbelgbt_1uge urgias • Sbeigbt_luge • 12 * 1111

// Calcule a1 el taufto de la fuente encaja y red6acalo baata que lo baga
// Parta del taullo -'• grande que panaca encajar ea lo• botoaea
Sfoat_•laa • 11;

// Deberla indica a GDZ d6nde •• alojan laa fueatea
putaav('Gl>rol!TPATII.C::\WIIIDOlfS\PoaU ')1
Sfoataaae • 'ar1al1

do
```



```
  $foat_•i• ··1

  //..._...el t-llo del tato en dlcbo t..-0 ds fuente
  $bbox•1.-gettfllboa !$foat_•1ae, o, $foatnmme, $11Uttoa_te~>!~

  $rigbt_tat • $bbox(Z11 // coordemda dsncba
  $left_tat • $1>boX[O11 // coordenada ia111lierda
  $Widtll tut • $rigbt text • $left text1// ¿cual•• su aclnara?.
  Sbeigllt_text • a1191$bllox17) • $bbox[1))1 // ¿cu61 •• su altura?

)
wbile C $foat_•iae>I ..
  C $beigbt_texc»$be1gbt_1uge •argiu ll
    $w1dtll_text>$Widtb_iuge_11O_urg1a• >
  );

if C $belgbt_text>$beigbt_iuge_110_urgiaa ll
  $Widtb_tut>Sw1dtb_1uge_11O_..rg1aa)

  // no encaja 11,iagGn taufto legible ea el botCID
  ecbo 'TeXt g1fta •U1 aot fit oa buttoa.cbr />';

el••
(

  // a- eDCODtrado un tamallo de fuente 1111• —•:J•
 .11 Abara da~ determinar d&lde colocarlo

  Stext_x • $w1dtb_i..ge/2.0 • $W1dtb_text/Z.01
  Stut_y • Sbeigbt_i11119e/2.o • $baigbt_text/2.0 1

  if C$left_tmtt e O)
      Stut_x +• ab• C$left_tutl, // agngue faccor para proyacci6n isquierda

  $aboft_line_text • abs C$bbox(71) 1 // ¿altura coa respecto a la U.nea baff?
 . Stext..y +• Saboye_liae_taxt, 1/ agregue factor de U.nea baH

  Stut_y ·• 21 // factor de ajuste para la f- de nuestra plantilla

  SWbit• • I•geColorAllocate CS1•, 255, 255, 255);

  ImegeTTl'Text 1$1•, Sfoat_•iae, O, $text_x, Stext_y, Swbita, Sfoataa ...
      $buttoa_tutl1

  ..ader !'Coateat-type, 1..ge/pag'1 ı
  Imgel'llg ISi•1 ı
```

Seguir al siguiente operativo

&ta es la secuencia de comandos mú larga que hemos examinado hasta ahora. Vamos a analizarla sección por sección. Comenzamos por UJUI sencilla comproba·ci6n de errores y establecemos el lienzo sobre el que comenzaremos a trabajar.

Determinar el lienzo base

En el listado 21.2. en lugar de empezar desde cero, partiremos de una imagen existente para el botón. Disponemos de tres opciones con respecto al color del botón: ro;o(red-button. png), verde (green-button. png) y azul (blue-button. png). El color seleccionado por el usuario se almacena en la variable $co;or desde el formulario.

En primer lugar extraemos el color de la variable superglobal $_RBQUBST y establecemos un nuevo identificador de imagen en función del botón pertinente:

```
$co~or • $_RBQUSST[•color•J1
$1•.  Imag.CreateFromPNG ($color.'button.png'l1
```

La función I-geCreatePromPNG () toma un nombre de archivo de una imagen PNG como parámetro y devuelve el nuevo identificador de imagen para la imagen que contiene una copia de dicho PNC. Tenga en cuenta que este hecho no modifica la imagen PNC base en absoluto. Podemos utilizar las funciones I-gecreatePromJPBG() y I-geCreatePromGIF() de la misma fonna que se ha instalado la compatibilidad adecuada.

Tipo	Tamaño	Espacio de almacenamiento (bytes)	Descripción

La llamada a ImageC:reaEePromPNG () sólo crea la imagen en memoria. Para guardar la imagen en un archivo o dirigirla al navegador para mostrarla, debemos llamar a la función ImagePNGf). No tardaremos en abordar este aspecto, pero todavla nos queda trabajo"qüe realizar' sobre nuestra imagen.

Ajustar el texto en el botón

En la variable $but to~ text tenemos almacenado texto escrito por el usuario. Nuestra intención es imprimir dicho texto sobre el botón utilizando el cuerpo de fuente mú grande que encaje en dicho espacio. Para ello, utilizamos el conocido método de ensayo y error mediante iteraciones. En primer lugar establecemos algunas variables relevantes. Las dos primeras son la altura y la anchura de la imagen del botón:

```
...._.._....,.• 1—geSX($im)1
au.-_1...,e  • I • gaST($iml;
```

la NgIII\du representan el margen desde el borde del botón. Las imágenes paaIG!f"Ntanes que estamos utilizando tienen relieve, por lo que tendremos que prnw-,.cio •Indedor de los bordes del texto. Si va a utilizar inuigenes düerentes, e. YIllor_led distinto. En nuestro caso utilizaremos un margen de 18 pfxeles a cada ladoJ


```
$width_image_wo_ • rgiD8• $width_image - (2 • 18)1
$height_image_wo_ • rgiD8• $height_image - (2 • 181
```

También necesitamos establecer el tamaño de fuente inicial. Comenzaremos por 32 (en realidad 33, que iremos i:edudendo) porque éste es, aproximadamente, el cuerpo de texto nuis grande que encaja en el botón:

```
$font_•ize • 33;
```

Si utiliza CD2, deberá indicar dónde se alojan las fuentes. Para ello, debe establecer la variable de entorno GDPONTPATH de la siguiente forma:

```
putenv('GDFONTPATR•C:\NIIID0lfS\Pont•'l;
```

También establecemos el nombre de la fuente que queremos utilizar. Vamos a utilizar una fuente con funciones TrueType, que buscará en el archivo de fuente situado en la ubicación anterior y adjuntará el nombre del archivo con . ttf (TrueType Font).

```
$fontname • 'arial';
```

Tenga en cuenta que dependiendo del sistema operativo puede que tenga que agregar . t tf al final del nombre de la fuente.

Si no dispone de la fuente Aria) (que vamos a utilizar aquí) en su sistema, puede sustituirla fácilmente por otra fuente TrueType.

A continuación,utüizanl0:5un bucle para ir i:educieildo el tamaño de la fuente hasta que el texto remitido encaje en el botón:

```
do
{
   $font_dze--1

   // determine el t&laailo del texto corre•pondiente a dicho cuerpo de texto
   $bbox•i•gettfbllox ($font_•ize. 0, $fontna... $button_text);

   $right_text • $bbox(2J;                    // coordenada de la derecha
   $left_text • $bllox(0);                    // coordenada de la izquierda
   $width_text • $right_text   $left_text,    //¿~le•  •u anchura?
   $height_text • ab•($bbox(71  - $bbox(11);  //¿~le•  •u altura?


vhile ( $font •ize>8 ••
       ( $height_tezt>$height_i•ge_wo_•rgiu 11
         $width_text>$width_image_wo_•rgiD81
   );
```

Esta secuencia de código comprueba el tamaño del texto examinando la caja del texto. Para ello, utilizamos la función ImageGetTTPBBox(), que es una de las funciones de fuentes TrueType. Tras averiguar el tamaño del texto, lo imprimimos sobre el botón utilizando una fuente TrueType (en este caso Aria), pero puede utilizar cualquier otra que desee) y la función I-geTTFText (). El cuadro delimitador de una secuencia de texto es el recuadro nuis pequeño que se puede

EL_HOLISTICO@HOTMAIL.COM - 2006

dibujar alrededor del texto. En la figura 21.6, se incluye un ejemplo de un cuadro delimitador.

~urCompan~

Figura 21.1. Las coonleMdU del cuadro dellmltador ae establecen en funcl6n de la llnea baae. El origen de la coordenadas utilizado aqul es (0,0).

Para obtener las dimensiones del cuadro, llamamos a

```
$bbox•i.agettfbllox ($font_aiae, 0, $fontna .., $button_text);
```

Esta llamada se interpreta de la siguiente forma: "para el tamafto de fuente $f ont _81 ze, con el texto inclinado a un útgulo de 0 grados y utilizando la fuente Arial, indicar las dimensiones del texto en $button_text".

Tenga en cuenta que nec:;esitamospasar la ruta al archivo que contiene la fuente dentro de la función. En este caso, se trata del mismo directorio que la secuencia de comandos (el predeterminado), por lo que no se ha especificado una ruta m4s larga.

La función devuelve una matriz con las coordenadas correspondientes a las esquinas del cuadro delimitador. En la tabla 21.1 se recogen los contenidos de la matriz.

Tallll 21.1. Contenidos de la matriz del cuadro dellmltador.

Tipo especificado	Formato de visualización
0	Coordenada X. esquina Inferior izquierda
	Coordenada Y, esquina Inferior izquierda
2	Coordenada X, esquina Inferior der9cha
3	Coordenada Y, esquina ~ der9cba
"	Coordenada X. esquina superior derecha
5	Coordenada Y, esquina superior derecha
6	Coordenada X, esquina superior izquierda
7	Coordenada Y, esquina superior izquierda

Para recordar los contenidos de la matriz, basta con tener en cuenta que la numeración comienza en la esquina inferir izquierda del cuadro delimitador y va aumll!ntando en el sentido de las agujas del reloj.

Es necesario tener en cuenta una característica especial relativa a los valores de la función 1—g•TTPBBoxc): se trata de coordenadas especificadas desde un origen; sin ~bargo, a di~ de las coordenadas utilizadas con ilúgenes, que se

especifican en función de la esquina superior izquierda,. en este cuo se hace en función de una linea base.

Vuelva a examinar la figura 21.6. Como observan, se ha dibujado una línea que subraya el texto. Esta lfnea es lo que se conoce como linea base. Algunas letras quedan por ~jo de esta lfnea-base, como la "y". &tas se conocen como descendentes.

La parte izquierda.de la línea base se utiliza como origen de las dimensiones, es decir, las coordenadas X eY serán 0. El valor de las coordenadas Xpor encima de la línea base serA positivo y por debajo serAnegativo.

Adem4s, el texto puede tener valores de coordenadasconespondientes a puntos situados fuera de la caja. Por ejemplo, el texto podria empezar en la coordenada X con valor -1. Por lo tanto, debemos tener cuidado a la hora de realizar cüculos con estos números.

En nuestro caso, vamos a calcular la anchura y la altura del texto de la siguiente forma:

```
$right_text • $bbox[2)1                    ti coordenada derecha
$left_text • $bbox[OJ1                     11 coordenada iaquiarda
$widtb_text • $right_text • $left_text1    ti ,CUal ea la ancbllra?
$heigbt_text. eba($bbox[1) • $bbox[1));    11 ¿cull ea la eltura7 •
```

Tras ello, probamos la condición del bucle:

```
} while csfont aiae>8 ,,
    ...e• $Miglae_~$lulight_imqa_wv..rgiu 11
        $widtb_text>$widtb_i..ge_wo_..rgina !
      )i
```

En esta secuencia se prueban dos conjuntos de condiciones. El primero es que la fuente resulte legible (no es aconsejable utilizar fuentes con un tamaAo inferior a los 8 puntos porque el botón resulta difícil de leer). El segundo conjunto de condiciones comprueba si el texto encajarAen el espacio que le hemos asignado.

A continuación. comprobamos si los cAlculos del proceso de iteraciónobtienen un tamafto de fuente aceptable o no; en caso negativo, devolvenn un error:

```
if e $beigbt_text>$height_i..ge~wo—•rgina 11
     $widtb_text>$width_i•ge_wo_•rgina)

   11 ning6n t...llo legible encaja an el bot&l
   echo 'Text given will not fit on button.cbr />';
```

Colocar el texto

Si todo sale bien, el siguiente paso consiste en calcular el punto en el que colocar la parte iniáal del texto. Se trata del punto medio del espado disponible.

```
$text_x • $widtb_image/2.D • $widtb_text/2.D;
$text_y • $beigbt_i•ge/2.D • $beigbt_text/2.0;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Debido a las complicacionescon la linea bue relativa al sistema de coordenadu, necesitamos agregar algunos factons de correcci6n:

```
if 1$left text e 01
   $text_i' +• ab•IS1•ft_textl;. // a9regue factor para prayecci6a izquierda

$above liae text • ab•ISbb0x1711; 11 ¿Hpacio por eaci- de la Uaea tlaH?
$text_Y +• 'iatiove_liae_text;     // agregu• factor para la Uaaa baH

Stext_Y -• 2; // factor de aju•te para le forma de aueetra plantilla
```

Estos factores de correcciónpermitenrealizarunos pequeftosajustes para corregir la altura de nuestra imagen.

Escribir el texto en el botón

Tras ello, el resto es sencillo. Establecemos el color del texto, que ser6 blanco:

```
$wbite • 1-9eColorA1locat• ($1•, 255, 255, 2551,
```

Podemos utilizar la función ImageTTFText () para dibujar el texto en el botón:

```
Iaage'l'TPText 1$ia, $foat_•iz•, 0, $text_x, $text_y, $white, $fontaa..,
$buttoa_text);
```

Esta función toma bastantes parámetros. Siguiendo su orden, se trata del idei.itificador de la imagen, el tamaño de la fuente en puntos, el Angulo al que dibujar el texto, las coordenadasX e Y del texto, el color del texto, el archivo de la fuen~ y, por último, e~ texto en sf.

Para terminar

Por último, podemos generarel botónen el navegador:

```
Beader l'Conteat-type, iaage/png• 11
I11a9ePDg !Sial,
```

Seguidamente, llega el momento d!i! liberar los recursos y finalizar la secuencia de colllUldos:

```
ImageDestroy ($im);
```


Y eso es todo. Como resultado, debería aparecer un botón en la ventana del navegadorsimilar al que se ve en la figura 21.S.

Dibujar figuras y representación gráfica de datos

En la última aplicación, trabajamos con imAgenes y texto existen•. En el siguiente ejemplo veremos cómo realizar dibujos. A continuación, desarrollaremos una encuesta en nuestro sitio Web para comprobar a quWn votarían los usuarios en unas elecciones ficticias. Almacenaremos los resultados de la encuesta en una bue de datos MySQLy ch'bujaremos un gnfico de barras con dichos resultadosutilizando funciones de imagen. Estas funciones también se utilizan para crear gr6ficos. Puede generar diagramas con los datos deseados: ventas, visitas a la Web, etc.

En este ejemplo,configuraremos una base de datos llamada po11. Esta base de datos contiene una tabla llamada poll_reeulta, que incluye los nombres de los candidatos en la columna candidate y el nwnero de votos recibidos en la columna nu•_votee. Tambim hemoe erado un usuario para esta bue de datos llamado pol 1 cuya contraseña es poll. La operacióncompleta de configuraciónlleva unos cinco minutos si ejecuta la secuencia de comandos SQL ilustrada en el listado 21.3. Para ello puede dirigir la secuencia de comandosa travá de un inicio de sesión de usuario raíz de la siguiente forma:

```
,ay•ql -u root -pe poll•etup.•ql
```

También puede utilizar el inido de sesión de cualquier otro usuario que disponga de los pri~legios MySQL apropiados.

Llsmdo 21.3. pollsetup.sql. Cruc:i6n de lll bHe de d•to• Poll.

```
create databa•• poll;
-  poll;
create table poll re•ult•
   candidate vareBar(10I,
   D__ YOte• iDt
);
inaert iDto poll_reeulta value•
   (•Jobn Sllith', 0).
   l'Mary J-e•, O),
   l'Fred llogg••, 0)
;
grant all privilege•
oa poll.•
to pollelocalho•t
identified by 'poll•;
```

Esta base de datos contiene tres candidatos. Creamnos una interfaz de voto a trav6s de una página llamada vote.html. En el listado 21.4 se incluye el código para esta página.

EL_HOLISTICO@HOTMAIL.COM - 2006

Ustado 21.4. vote.html. Los usuarios pueden votar a través de esta pégina.

```
cbtml>
cbead>
   ctitla>Pollingc/titla>
cbaad>
cbody>
cbl>Pop Pollc/bl>
cplltbo will you vota for in tba alaction?c/p>
cfora ..tbod•"poat• action••abow_poll.pbp">
cinput typa••radio• na •vota• valua••John Smitb">John Smitbcbr />
cinput type••radio• na •vota• valua••Mary Jonee">Mary Joneacbr />
cinput type••radio• na•a••vota• value.•rred Blogga">Fred Bloggacbr />cbr />
cinput typa••aubllit• valua••Sbow raaulta">
c/form>
c/body>
cbtml>
```

El resultado de esta página se muestra en la figura 21.7.

Figura 21.7. Los usuarios pueden votar y, si hacen elle sobnt el bol6n de envio, se mosnrén los resultados acluaJes de la encuesta.

Cuando los usuarios hacen clic sobre el botón, agregaremos sus votos a la base de datos, obtendremos todos los votos de la base de datos y dibujaremos un diagrama de barras con los resultados actuales.

En la figura 21.8 se recoge un diagrama típico tras recoger varios votos.

La secuencia de comandos que genera esta imagen es bastante larga. Vamos a dividirla en cuatro partes que analizaremos por separado. La mayor parte de la secuencia de comandos resulta conocida ya que hemos examinado bastantes ejemplos de MySQL parecidos. Hemos visto cómo pintar un lienzo de fondo en un color sólido y cómo imprimir las etiquetas de texto sobre él.

Figura 21.1. Los resultados de los votos se crean dibujando sertas de llneas, recténgulos y elementos de texto sobre un lienzo.

Las partes nuevas de esta secuencia de comandos son las encujadas de dibujar líneas y rectmgulos. Nos centraremos en estas secciones. En el listado 21.5.1, se recoge la parte 1 (de las cuatro partes que componen la secuenciade comandos).

u.tNo 21.5.1. show_poll.php. La parte 1 actualiza la base de datos de votos y recupera los nuevos resultados.

```
c?pbp
}
   Conaulta para obtener informaci6D
.........................................}

//"obte119ae1 voto deade el formulario
$vote-$_11SQUBST('vota•J1

// inicie aeai6n en la baae de dato•
if (!$clb _conn • new 11}'aqlil'localboet•, •poll', 'poll', 'poll'II
(
   ecbo 'could not connect to dbcbr />';
   exit;


if (!empty(Svotell  // ai el uauario rellana al foraulario agregue au voto
(
   $vote• addalaabea($vote);
   Squery • •úpdate poll_reaulta
            aet num votea • num votes+ 1
            where candidata. •ivote";
   if(I!CSreault • eSdb_conn->query($queryllll
   (
      echo •could not connect to dbcbr />';
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
      ea.t1

// obtenga Ío• re•ultado• actual•• ele la enc:ue•ca, bayan YOtado o -
Squary • '••lec:t: • fr-  poll_re•ulu '1
if ⁰I C$neulC • e$db_conn-:,,queryCSqueryl 11
(
   ecbo  •c:ould not connec:t to dbcbr  />';
   eait1
)
sn-_caad:1.datu  • -r-ql_n__ —— csre-ltl;

// calcule al 1111..ro total de votoa baata al ID011ento
Stotal vot•••01
vbile csrov. $reault-..fetcb_object111
(
      Stotal_vot••  +•   srov-..111111_vote•1
)
Sr••ult·>data_aeek(0I;  //  reatableaca el puDtaro de reaultaclo
```

La parte 1, ilustrada en el listado 21.S.1, establece la conexión a la bue de datos MySQL, actualiza loe votos en función del usuario introducido y recoge loe nuevos votos.

Tras obtener dicha informadón, podemos empezar con los álculos para dibujar el gñfico. En el listado 21.S.2 1e muestra la parte 2.

—ií62'f."5%"1ffiMpolf.plIi:l.ipiili Zutilirec:4ftocfii1 Iiii viirimrN para ill dibujo.

```
/•.....%.....................................
  c:Alculo• inicial••  paraalgr,fico
..'''..''.'....'.......'..............'...   ,
// aetableaca laa con.tantea
p11ten,r(ıQDIPOIITPATH•C::\WIIIDOIIS\Ponta'11
Svidtb•500;    // ancbura de la :1.. gen an plzelaa - ,ata encajar, an &4011480
Slaft_..rgin • 50;   // aapacio que dejar a la izquierda de la i-gen
Srigbt_..rgiD• 501   // upacio que dejar a la darecba da la iugan
Sbar_baigbt • 40;
Sbar_apacing • Sbar_beigbt/2;
Sfont • 'arial';
Stitle_ai••• 1&1  // puDto
s..ia_•ize• 12, // punto
s.-1I_aiz•• 12, // pUDto
Staat_indeDt • 101//  ubicaci6D para laa etiqueta• da tasto a la izquierda

// eatablezca el puato iDicial de•de el qua dibujar
Sa • Sleft_..rgin + &0;   //  lugar en al que dibujar la lilwa baae dal
//gr&f:l.co
Sy • 50;        // lugar en el que dibujar la llnea b••• dal gr,fico
Sbar_'llllit •  ($Vidtb- ($z+Srigbt_..zginl 1 / 100,    // UD "Jlllllto• en el gr&fico

// calcule1a  altura dal gdfico - la• barra•, ú•  lo• lmec:oe, 116• cierta
// di•tancia da ..rgea
Sbaigbt: • s—_candidatu • ISbar_baigbt + $bar_epaciagl + 50;
```


La parte 2 establece algunas variables que se pueden utilizar para dibujar el gr6fico.

Elálc:ulo de los valores para este tipo de variables puede nsultar tedioso, pero si 1e prev6 con antelación el aspecto que desea que tenga la imagen al final el proceso de dibujo n!llltar6 mucho mú sencillo. Para obtener los valora utilizados, dibujamos el efecto deseado en un papel y estimamos las proporciones necesarias.

La variable $width es la anchura total del lienzo que utilizaremos. Tambim establecemos 101 m6rgenes izquierdo y derecho (con $left_margin y $rigbt_margin); el grosor y el espacio entre las barras ($bar_height y $bar_apac:ing); y la fuente, los tamaftos de fuente y la ubicadón de las leyendas ($font, $title_sh:e, $main_size, $small_size y $text_indent).

Partiendo de estos valores; podernos realizar unos cuantos ailculos. Queremos dibujar una linea ba1e desde la que partan todas las barras. Podemos determinar la ubicación de esta linea ba1e utilizando el margen izquierdo mú un espado para las etiquetas de texto para establecer la coordenada X y un valor estimado obtenido de nuestro boceto en papel para la coordenada Y.

Tambim podemos calcular dos valores importantes: en primer lugar, la distancia en el gdfico que representa una unidad:

```
$bar_UDit • ISvidtb·CSa+Sr:1.gbt_marginl1   / 1001   // 11D •punto•  en el  gr6f1co
```

Se trata de la longitud m6xima de las barras (desde la lfnea bue hasta el margen dereehe) dMdiclo por 199 porque uuesbo gxáfico va a nabar vidures de porcentaje. El segundo valor es la altura total que necesitaDIOS para el lienzo:

```
$beig bt • s—_candidatea • isbar_bei9bt + Sbar..,.•P.9ciD91 + .50;
```

Se trata b6sicamente de la altura de la barras multiplicada por el número de barras, más una cantidad extra de espado para el tftulo. En el listado 21.S.3 se recoge el c6digo correspondiente a la parte 3.

u.t.do 21.5.3. lhow_pall.php. la pmrllt 3 CIU el gr6fk:o y lo d9ja llsl0 para agregar datos.

```
,.......'.......''...'.........................
   Crearla  i..9en de baH
.....''.....''...'..'.....'''.....'....,
// Cree un lienao vaclo
Si••  I-v&CreateTrueColor(Svidtb,$bei9bt1;

// Aai9119 color••
$vbite•ImageC:OlorAllocatelSia,255,255,255J;
Sblue•Illagec:olorAllocate(Sia,0,&4,1211;
$black•laageC:olorAllocatelSi•,0,0,011
$p:I.Dk • 1..gac:olorAllocate(Sia,255,71,2431;

Stezt color•  Sblack;
$perceat_color • Sbl"ack;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
$bg_color • $wbite,
$line_color • $black1
$bar_color • $blue,
$nUllber_color • $pint,

// Cree el lienso en el que dibujar
1..gePilledRectánglel$ia,0,0,$widtb,$heigbt,$bg_colorl1

// Dibuje llneaa alrededor del lieaso
1..geRectaaglel$ia,0,O,$width-l,$baigbt-l,$liae_colorl1

// Agregue el titulo
$title • •,011 aeaulta';
$title_d1..aeioaa • 1..geTTPIUlox1$title_eise, o, $font, $title);
$title_length • $title_di..naional21 - $title_di..naionalOJ,
$title_beigbt • abe1$ticle_di..naional7J - $title_di..naional1J);
$title_above_line • .tla1$title d1..naiona[7JI;
$title:x • 1$width-$title_length)/2; // clncrelo en x
$title_y • ($y - $title_baigbtl/2 + $title_above_line; //clntrelo en y
1..geTTFTextl$ia, $title_aise, 0, $title_x, $title_y,
      $text_color, $foat, $titlel;

// Dibuje una Unea bHe ligeraaente por enci.. de la pri..ra barra
// baaca un poco por debajo de la 6ltl..
1..geLinel$i•, Sx, $y-s, $x, $height-1s, $11ne_colorl,
```

En la parte 3 creamos la imagen de base, asignamos los colores *y* comenzamos a dibuja, el grfflco·Ptlre relleMr el fondo del pfico utilizamos·

```
ImageP1lledRectangle($1a,0,0,$widtb,$height,$bg_colorl;
```

La función ImageFilledRectangle (), como podría imaginar, dibuja un rect.ingulo relleno de color. El primer parámetro es, de la fonna habitual, el identificador de la ianagen.

A continuación, debemos pasar las coordenadas X e Y del punto inicial *y* del punto final del rect6ngulo. &tas se corresponden con la esquina superior izquierda *y* con la esquina inferior derecha, respectivamente. jn este caso, rellenamos el lienzo completo con el color de fondo, que es el último parúnetro *y* es blanco. A continuadón llamamos a

```
1..geRectangle($ia,0,0,$width-l,$helght-l,$line_colorl,
```

para dibujar un línea negra por el borde del lienzo. Esta fundón dibuja un rect.ingulo sin relleno. Los par4metros son los mismos. Fíjese en que hemos dibujado el rectllngulo con las medidas $widtb-1 *y* $beigbt-1. Como resultado, se crearll un lienzo con una altura *y* achura según estos valores desde (0,0). Si lo dibujamos con los valores $widt;b *y* $beigbt, el rectllngulo saldrá fuera del llrea del lienzo.

Utilizamos la misma lógica *y* las mis.nas funciones que en la últiana secuenda de comandos *y* escribimos el tftulo en el gráfico. Por último, dibujamos una línea base para las barras con

```
I..geL1ne($ia, Sx, $y-s, $~, $heigbt·15, $11ne_colorl;
```


La fundón I—geLine () dibuja una lfnea en la itnagen espedficada ($ia) desde un conjunto de coordenadas '($x, $y-S) a otro ($x, $beigbt-15), en el color especificado por $line_color.

En este caso, cb"bujamos la lfnea base ligeramente por endina de dónde queremos dibujar la primera barra hasta m1 poco por encima de la parte inferb del lienzo. Ya estamos preparados para aeUenar de datos el grllfico. En el listado 21.S.4 se muestra la parte 4.

Listado 21.5.4. lhowpoll.php. La parte 4 dibuja lol dalol reales IOln el grMco '/ termina.

```
,...'.................'....'...'........'....
  Dibujar datoa en el grifico
....'.................'........'........'...  ,
11 Obteng• cada Unea de la baN de datoa y dibuje laa barraa correapoadieatea
while ($row • $reault->fetcb _object(I)
(
  if ($tocal votea > 0I
    $percent • intvd ( ($raw->-  wtea/$total wte•I •1001l,
  dn
    $percent • 0;

  // aueatre el porcentaje de eate valor                    ~~
  $percent_diaenaiona • I11a99Trr811ox($alein_aise, o, $font, $percent.'1'11
  $percent_length • $percent_diaeneiona[21 • $percent_di..nai9D9(011
  ~$.la.              $M.la.._ai.u. .. D.. lnj~b -Jp ! ~'!D!; ~'!1!ftb~J':~~-indent,
      $y+ ($bar_heigbc/21 , $parcent_color, $font, $percent. '1•1;

  // longitud de la barra para eate valor                  .;
  $bar_lengtb • Sx + ($parcent • $bar_unitl1

  // dibuje la barra para eate valor
  1..gerilledaectangle1$ia, $x, $y·2, $bar_lengtb, $y+$bar_beight, $bar_colorl;

  //.dibuje el cltulo para eete valor
  ImageTTPTexcl$1•, $..in_•i•e, 0, $text_indent, $y+1$bar_height/21,
              $text_color, $font, $row·>e•ndidatel;

  // dibuje la linea aoatrando el 1DOI
  I..geReccangle($1a, $bar_length+1, $y-2,
               ($x+(100*$bar_unitl1, $y+$bar_beigbt, $11ne_colorl,

  // 1111eacre loa nGaeroa
  I..geTTPText($1a, S... 11_aise, 0, $x+(100*$bar_un1tl-SO,
              $y+($bar_height/21,
               Snllllber_color, $foat, Sraw->nua_votea.•/•.$tot•l_voteal;

  // P••e al• •iguiente b•rra
  $y•$Y+1$bar_height+$bar_apaclngl;

,..'........'............'...'.............
  Noatrar 1..gen
```

```
.....'.....'..'....   ,
Header1 •coateat-type,  image/pnv' 1;
ImagePng ($ia1 ;

,.....'.'....
   Liberar rec:url1e1•
.....'.....'..'...'.'.'''.   ,
1.. geoeatroy($ia1;
?>
```

La parte 4 recorre los candidatos desde la base de datos uno • uno, calcula los porcentajes de votos y dibuja lu bllrru y las leyendas pan cada uno. Para agregar las leyendas hemos vuelto • utilizar la función I-geTTF'l'ext t}. Dibujamos las barras como rect4ngulos rellenos de color utilizando ImageFilledRectangle 1) :

```
I • gel'illedaectugle($ia, $x, $y-2, $bar_leagth, $y+$bar_beight, $bar_colorl1
```

Agregamos los contornos para indicar el 100% utilizando I-geRectangle ():

```
1.. 9eRectaaglel$ia, $ber_leagth+1, $y-2,
        ($X+1100•$bar_UDItl1, $y+$bar_height, $liae_colorl,
```

Tras dibujar las barras, volvemos • generar la imagen con I-gePRG () y liberamos los recursos con I-geDeatroy (>.

&ta s.ecuenc:ia de comandos es larga, pero resulta fkil de adaptar a las dilemates neceactades a a las enniestas autageneradas 1 tnvAs de .,.,. iotemz Un~ mento importante que falta en esta sec:uenc:ia de comandos es 1U1 mecanismo contra engai\os ya que los usuarios no tardar6n en descubrir que pueden votar tantas veces como deseen, lo que har6 que losresultadoa no sirvan para mucho•.

Puede utilizu el mismo método para dibujar gr6ficos de lineas o incluso gr66-c:os cin:ulares,si se le dan bien las matem6ticu.

Otras funciones de imagen

Adem6s de las funciones de imagen que hemos utilizado en este capítulo, existen muchas otru. Para dibujar por medio de un lenguaje de programac:i6n se necesita tiempo y realizar muchas pruebas hasta conseguir el resultado correcto. Comience siempre con un boceto de lo que desee dibujar y, a continuac:i6n,consulte el manual par• buscar cualquier función que pueda necesitar.

Lecturas adicionales

En Internet encontrar, 1UI8 gran cantidad de material de lectura. Si tiene problemas con las funciones de imagen, a veces resulta aconsejable examinar la documen-

taci6n de GD penque las funciones de PHP c:onties,m esta biblioteca. Para ello, dirfjase a bttp: //wwv. boutell . ca./gd.

Recuerde que la versión de GD2 pua PHP es una vuiante de la biblioteca principal, por lo que alguno» detalles pueden variar. Tambim existm algunoe cursos pr6c:t:ic:os excelentee eobre tipos dadoe de aplic:ac:ioNsgrfflcu,en coacreto en Zend y Devshed que puede encontrar en bt tp: //wwv.zend. com y bttp: //devabed• com,. respectivamente.

La aplicación de diagrama de barras utilizada en este capftulo •t6 inspirada en la aec:uenc:ia de comancloa de diagrama de burM dimmicoelCl'ita parSteve Maranda y disponible en Devshed.

A continuación

En el siguiente capítulo, analizaremoe la pr6ctic:a fund~d del control de sesiones de PHP.

EL_HOLISTICO@HOTMAIL.COM - 2006

22

Utilizar sesiones

En este capftuJo anaJlzamnos la funcionalidad del control de sesiones en PHP. Nos centraremos en los siguientes aspectos:

- Concepto del control de sesiones
- Cookies
- Configurar una sesión
- Variables de sesión
- Sesiones y autenticación

Concepto del control de sesiones

Seguramente habrá escuchado que H1TP es un protocolo sin estado. Esto significa que el protocolo no dispone de un método incorporado para conservar el estado entre dos transacciones. Cuando un usuario solidta una p4gina y después otra, H1TP no cuenta c~n ninguna ticnica para que podamos saber que ambas solicitudes provienen del mismo usuario.

El concepto del control de sesiones consiste en poder realizar el seguimiento del usuario durante una sola sesión en un sitio Web. Si podemos hacerlo, podremos conectar a un usuario y mostrar contenido en función de su nivel de autorización o de sus preferendu personales. Podemos realizar el seguimiento del comporta-

EL_HOLISTICO@HOTMAIL.COM - 2006

miento del usuario. Podemos implementar un carro de la compra. Desde la venión 4, PHP incluye funciones de control de sesiones propias. El enfoque del control de sesiones ha variado ligeramente con la aparición de las variables superglobales; ahora podemos utilizar la variable global $_SBSSION.

Funcionalidad básica de sesiones

En PHP, las sesiones se controlan por un solo Id. de sesión, un número criptognificamente aleatorio, generado por PHP y que se almacena en el lad~ del cliente mientras dura la sesión. Se puede almacenar en el ordenador del usuano en forma de cookie o pasarse por medio de URL.

El Id. de sesión actúa como una clave que le permite registrar determinadas variables como variables de sesión. Los contenidos de estas variables se almacenan en el servidor. El Id. de sesión es la única información visible en el lado del cliente. Si al realizar una determinada conexión a su sitio, se puede ver el Id. de sesión por m~io de una cookie o del URL, podni acceder a las variables de sesión almacenadas en el servidor (aunque puede cambiarlo para utilizar una base de datos si pretende escribir su propia función, como veremos en el apartado sobre configuración del control de sesiones).

Probablemente habni utilizado sitios Web que almacenen un Id. de sesión en el URL. Si en su URL hay una cadena de datos aparentemente aleatorios, es muy probable que se trate de alguna forma de control de sesión.

Las cookies son una solución düerente al problema de conservación del estado entre varias transacciones al tiempo qu~ se mantiene un URL de aspecto limpio.

Definición de cookie

Una cookie es un pequeño fragmento de información ~ue las secuen~s ~e comandos pueden almacenar en un equipo del lado del cliente. Puede definir una cookie en el equipo de un usuario si envía un encabezado HTrP que contenga datos con el siguiente formato:

```
Set·Cookie,    .NONJIRB•VAI.OR|   [expires•P.ICIL\¡)   [path•Rln'A;)
  [4-ain•.NONJIU_DOIfJ1fEO[jHcurel
```

De esta forma se crea una cookie con el nombre NOMBRE y con el valor VALOR. El resto de panimetros son opcionales. El campo expires establ~ la fecha h:'5ta la que será válida la cookie (si no se establece una fecha de caducidad, la cookie será permanente a menos que la eliminemos manualmente). path y domain se ~~eden utilizar de forma conjunta para especificar el o los URL para los ~ue se utiliza la cookic La palabra clave secure significa que la cookie no se enviará a través de una conexión HTIP. Cuando un navegador se conecta a un URL, primero busca las cookies almacenadas localmente. Si ninguna de ellas es relevante para el URL al que se ha conectado, se devuelven al servidor.

Configurar cookies desde HTTP

Podemos configurar cookies manualmente en PHP por medio de la función setcookie(). Su sintaxis es la siguiente:

```
in~ Htcookie (string .aa.bre [, string valor [, int aducidad [, st:dng rvc:a
C. ·string doei.nio t. int Hcurel 11111
```

Los panimetros se corresponden exactamente a los del encabezado Set -cookie mencionado anteriormente. Si define una cookie como

```
••tcookie ( •,aycookie•, •value' 1;
```

cuando el usuario visite la siguiente página de su sitio (o vuelva a cargar la página actual), podremos acceder a la cookie a través de $_COOKIB¢ · mycookie · J o de $HTTP_COOICIB/ARS [•mycookie•J (o, si haactivado register_globala, directamente como $mycookie).

Se puede eliminar una cookie si se invoca setcookie () de nuevo con el mismo nombre de cookie y una fecha de caducidad pasada. También se puede definir manualmente una cookie con ayuda de la función header () y la sintaxis de cookie que mostramos anteriormente. Debe saber que es necesario enviar los encabezados de c:ookieantes que ningún otro encabezado o no funcionarán (se trata de una limitación de las cookies más que una limitación de PHP).

Utilizar cookies con sesiones

Las cookies tienen algunos problemas asociados como por ejemplo que algunos navegadores no las aceptan o que algunos usuarios pueden desactivarlas en isus navegadores. Es una de las razones por las que las sesiones PHP utilizan un método cookie/URL dual (como veremos en breve). Al utilizar sesiones PHP, no tendremos que definir las cookies manualmente. Las funciones de sesión se encargan de ello por nosotros.

Podemos utilizar la función session_get_cookie_parama () para ver los contenidos de una cookie definida por el control de sesión. Devuelve una matriz asociativa que contiene los elementos lifetime, path, domain y secure.

También podemos utilizar

```
••••ion_set_cookie_para .. ($lifetime, $path, $dollain [, $secure1);
```

para definir los parámetros de la cookie de sesión.

Si necesita más información sobre cookies, puede consultar la especificación sobre cookies en el sitio de Netscape, http: //home. netscape. com/newaref / std/cookie spec.html.

(Puede ignorar el hecho de que este documento se denomina "especificación preliminar", ya que l!a sido así desde 1995 y es lo más parecido a un estándar sin que reciba esta denominación.)

EL_HOLISTICO@HOTMAIL.COM - 2006

Almacenar el Id. de sesión

De forma predeterminada, PHP utiliza cookies con sesiones. Si existe la posibilidad, se..definirá una cookie para almacenar.el Id. de sesión.

El otro método que se puede utilizar consiste en ai\adir el Id. de sesión al URL. Podemos indicar que se realice automáticamente si configuramos la directiva session. use..;.trans-sid en el archivo php. ini, quede forma predeterminada está desactivada.

También podemos incrustar manualmente el Id. de sesión en enlaces para que se pase de esta forma. El Id. de sesión se almacena en la constante SIO. Para pasarlo manualmente, debe ai\adirlo al final de un enlace similar a un parámetro GBT:

```
cA RRBP•"link.php?c?php•cho atrip_taga(SIDl1 ?>°>
```

(En este caso se utiliza la función strip_ tags (I para evitarataques de secuencias de comandos entre sitios.)

Normalmente resulta más sencillo compilar con - -enable- trans - • id,. siempre que sea posible.

Implementar sesiones simples

-- Los pasos básicos del uso de sesiones son los siguienles:

- Iniciar una sesión
- Registrar variables de sesión
- Utilizar variables de sesión
- Anular las variables registradas y eliminar la sesión

Estos pasos no siempre tienen lugar al mismo tiemr., en la misma secuencia de comandos y algunos de ellos pueden tener lugar en varias secuencias. A continuación analizaremos cada uno de los pasos individualmente.

Iniciar una sesión

Antes de poder utilizar la funcionalidad de sesiones, seri necesario iniciar una !lni,,n. Hay dos formas de hacerlo.

l...l primera, y la más sencilla, consiste en iniciar una secuencia de comandos con UN 11.am.ad.a a la función aesaion_start ():

```
••••1on_atar~1,
```

Esta funci~n co~prueba si ya hay un Id. de sesión actual. En caso contrario, creará uno. Si y.a e,uste ~I Id., básicamente carga las variables de sesión registradas

para que podamos utilizarlas.~ acaasejable invocar session_start e) al inicio de lodas las secuencias de comandos que utilicen sesiones•

La segunda forma de iniciar una sesión consiste en configurar PHP para que lo haga automáticamente cuando alguien visite el sitio. Puede hacerlo por medio de la opción seas ion. auto_start en el archivo php. ini, que comentaremos alanalizar la configuración.

Este mélodo presenta un inconveniente: al activar auto_start, no podemos utilizar objetos como variables de sesión.

Registrar variables de sesión

El registro de variables de sesión ha cambiado últimamente en PHP. En PHP 4.1, las variables de sesión se almacenaban en la matriz superglobal $_SBSSION y también en la antigua $HTTP_SBSSION_VARSLe recomendamos que utilice $_SBSSION. Para poder crear una variable de sesión, basta con definir un elemento en esta matriz, como se indica a continuación:

```
$_SBSSION[•myvar•J • 5;
```

Se realizará el seguimiento de la variable que acaba de crear hasta ~ ~lice la sesión o hasta que la anule manualmente.

Utilizar variables de sesión

Para poder introducir una variable de sesión en ámbito y utilizarla, primero debe iniciar una sesión por medio de aession_start () • Tras ello podrá acceder a la variable de través de la matriz superglobal $_SBSSION, como por ejemplo $ SBSSION[I myvar I] •

Si utiliza un objeto como variable de sesión, debe incluir la definición de clase antes de invocar session_start () para volver a cargar las variables i:le sesión. De esta forma PHP sabe cómo reconstruir el objeto de sesión.

Si ha activado register_globals, podr, acceder a lu variables de sesión de sus nombres cortos, como por ejemplo $myvar, aunque no es una práctica recomendable. Si ha activado register_globals, recuerde que una variable de sesión no se puede reemplazar con datosGBTo POST, una buena opción de seguridad pero que debe recordar al disei\ar el código.

Por otra parte, tendr, que prestar especial atención cuando compruebe si se han configurado las variables de sesión (por ejemplo a través de i sset () o empt y e)). Recuerde que las variables las puede configurar el usuario a través deGBTo POST. Puede comprobar una variable para ver si se trata de una variable de sesión registrada por medio de$ _SBSSION.

Puede comprobarlo directamente si utiliza lo siguiente:

```
if (1•••tl$_SBSSION('iayvar•JJI •••
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Anular el registro de variables y eliminar la sesión

Cuando haya terminado con una variable de sesión puede anular su registro. Puede hacerlo directamente si anula el registro del correspondiente elemento de la matriz $_SESSION, como se indica a continuación:

```
unaet(S_SBSSlOll('myvar'J);
```

Sepa que el uso de session_unregister () y session_unset () ya no es necesario, ni recomendable. Estas funciones se utilizaban an~ de la aparidón de $ SESSION.

-No intente anular la matriz $_SESSION completa; ya que si lo hace podría deshabilitar las sesiones. Para anular el registro de todas las variables, utilice:

```
$_SBSSlOlf • array(I;
```

Cuando haya terminado con una sesión, primero deberla anular el registro de todas las varia!;>les y, tras ello, invocar

```
aeaaiOD_deatroy();
```

para borrar el Id. de sesión.

Crear Uf.I sencillo ejemplo de sesión

Seguramente todo esto le parezca ligeramente abstracto, por lo que yeremos un ejemplo. Implementaremos un conjunto de tres páginas.

En la primera página, iniciaremos una sesión y registraremos la variable $ SESSION [• sess var • J • El código para realizar esta operación se muestra en efiístado 22.1 -

Listado 22.1. page1.php. Iniciar una sesión y crear una variable de sesión.

```
c?php
   ••••iOD_atart 11;

   s_sBSSlOlll'••••-var'J  • 'Hello world:';

   ecbo 'Tbe CODteDt of $_SBSSIOll(\'••••-var\'I ia '
        '$_SBSSlOlll'••••_var•J.'cbr />';
?>
ca hr•f••page2.php°>Hext pagec/a>
```

Hemos registrado la variable y definido su valor. El resultado de esta secuencia de comandos se muestra en la figura 22.1.

El valor final de la variable de la página es el que estará disponible en las páginas posteriores. Al final de la secuencia de comandos, se serializa o congela la

variable de sesión, hasta que se vuelva a cargar por medio de la siguiente llamada a session start ().

Por esta razón comenzaremos la siguiente secuencia de comandos con la invocación de session_start (). Lo mostramos en el listado 22.2.

Figura 22.1. Valor inicial de la v~ble de~ ~ IK!r I!VV 1.php.

Ustado 22.2. page2.php. Accedera una variable de sesión y anular su regislro.

```
c?php
   ••••ion_atart();

   ecbo 'Tba content of $ SBSSIOII(\'•••• var\•J ia '
        .$_SBSSIOll('Haa:-var•J. 'cbr />' ;-

   UD8et($_SBSSIOlll'••••_var•J);
?>
ca href••page3.pbp•,Hext pagec/a>
```

Después de invocar session_start (), la variable $_SESSION[• seas var • J está disponible con el valor almacenado previamente, como puede comprobar en la figura 22.2. Una vez utilizada la variable, anulamos su definición. La sesión sigue existiendo pero la variable $_ SESSION[• sess _var • J ya no es una variable registrada.

Por último, pasamos a la página page3 • php, la última secuencia de comandos de nuestro ejemplo. El correspondiente código se incluye en el listado 22.3.

Ustaclo 22.3. page3.php. Finallzaciónde la sesión.

```
c?php

   ••••ioa_atart();
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
echo 'i'be content of $_SBSS~OB[\'••••_var\'I   i• '
      .$_SBSSIOll['••••-var•l,'cbr     />';

••••ion_d••troy(l1
```

Figura 22.2. El valor de la vertable de sesión se ha pasado junto el Id. de sesión a pag2.php.

Como puede mmprobar en la figun 22.3, ya no tenemos acceso al valor de $_SBSSION[•sess_var•].

Figura 22.3. La variable de la qua hemos anulado el registro ya no 8Slé dlsponlble.

En algums versions de PHP anteriores a la 4.3• produda un error al intentar anular elemenlol de $11T'l'P_SRSSION:,.VARS o $_SBSSIOIJ. Si no puede anular elementos (es decir, si permanecen configurados), puede intentar utilizar sesa ion_unregiater C) para borrarestas variables.

Para finalizar inVOCMlCIS aesaion_dHtroyO para elhnmareltd. de la sesión.

Configurar el control de sesiones

Existe un conjunto de opciones de configuracicSn para sesiones que puede defi. nir en su archivo php. ini. Algunas de las mú litiles, así como una descripdón de las mismas, se incluyen en la tabla 22.1.

• Tabla 22.1. Opciones de conflgurac:16n de sesión.

Order processed at 13:54, 26th March

Your order is as follows:

••••ion.auto_•tart	0(delectlvadli)	lnlcllllllÜIIINeulomffcMllnl9.
••••ion.cactw_mq1ire	110	Dellne la dllracl6n de lil piglMI de INi6n enCICM,en minlml.
••••ion.cookie_dollain	ftldl	DamHoque.. canllgu111 en le coolde dlMIl6n.
••••ion.cookie_lifeti ..	0	Dlncl6nclelecaaldedlld.dlHll6n• elequlpodelUIUello.El VIllorpredeter-lllMdo,o, lndlcaqueclul'll..que• ciell9eln.vepdor.
••••ion.cookie_path	/	Rut. queM conftgu111 en i. coallle de Nli6ll.
••••ion.–	PBPUSID	NolllllledlleHll6nqueM..._COIIIO nombre di la c:oolde en el*- del ...,..,_
••••ion.••ve_hmuller		o.1118d6ndeMel.._...loedelalde le1Ni6n. PuedelndlclrqueNnlede une bele de [illegible] que _..._.,.....Func.ionH.
••••ion ••ave_path		Rule•lequeMIR11*-lold.aos de la MSl6n. De forme !MIgen8l'lll, ...-eoPNedo•--*'sir-- do '/ dellnldo par He•ion.eave_ bandler.
••••iOD,UH_cookie•	I(aclMdo)	Conlligur9 NIIoINII '*' que utlllcen cooldel en el ladodel c:llen19.

EL_HOLISTICO@HOTMAIL.COM - 2006

Implementar la autenticación con el control de sesiones

Por último, veremos un ejemplo más e1-borado sobre el uso del control de sesiones. Probablemente el uso más habitual del control de sesiones consiste en realizar el seguimiento de los usuarios una vez autenticados a través de un mecanismo de inicio de sesión. En este ejemplo combinaremos la autenticación de una base de datos MySQL con el uso de sesiones para conseguir esta funcionalidad. De esta forma asentaremos las bases del proyecto que analizaremos en un capítulo posterior sobre autenticación y personalización de usuarios, y que podremos volver a utilizar en otros proyectos.

El ejemplo cuenta con tres sencillas secuencias de comandos. La primera, authmain. php, incluye un formulario de inicio de sesión y de autenticación para miembros de nuestro sitio Web. La seguncia, members_only. php, muestra información solamente a los miembros que se hayan conectado correctamente. La tercera, logout. php, permite cerrar la sesión de un miembro.

Para comprender cómo funciona, .fíjese en la figura 22.4, la página inicial que muestra authmain. php.

Flguni 22.4. Como el usuario no ha iniciado sesión, se muestra una página de inicio de sesión.

Esta página permite al usuario conectarse. Si intenta acceder a la sección de miembros sin registrarse primero, obtendrá el mensaje que se muestra en la figura 22.5. Sin embargo, si el usuario se conecta primero (con el nombre de usuario testuser y la contraseña test123, como definimos en un capítulo anterior) y

tras ello intenta acceder a la página de miembros, obtendrá el resultado que mostramos en la figura 22.6.

Figura 22.5. Los usuarios no registrados no pueden acceder a los contenidos del sitio; recibirán este mensaje.

Figura 22.1. Después de conectarse, el usuario puede acceder a la zona de miembros.

Veamosel código de esta aplicación. La mayor parte del mismo se ~oncentra en authmain. php, que se muestra en el listado 22.4. Lo analizaremos paso a paso.

EL_HOLISTICO@HOTMAIL.COM - 2006

```
c?pbp
••••ioa_•tart(l1

if Ci•••tl$_P08T('u••ridJI   .. i•••tl$_POST('P•••vordJii
(
   // •1 el u•uario •ceba deiatentar  conect•r••
   $u•erid • $_P08T('u•eridJ1
   $pa••-rd • $_P08T('pa••-rd'];

   $db_conn • new 11Y9qli(localbo•t',    'vebautb',   •vebautb•,  'autb'li

   if (ay11qli_connect_errno()I  (
      echo •eonaection to databa•• failed:'  .my•qli_connect_error0 ,
      exit0;


   $query • '••lect  • froa autborised u•er11 '
            '•vbere na-•$u•erid' '-
            .' and pa••vord••ba1('$pa••-rd'I";

   $result  • $db_conn•>query($query)1
   if  ($re•ult•>nwa_rov•  >0 l
   (
      // d  ••tAn en la ba•• da dato•, regbtre el Id. de u•uario
      $_SRSSIO11[valid_u••r'I   • su..rid;
   }
   $db_conn•>C108e(l1


?>
cbtml>
cbody>
cbl>llome pagec/bl>
c?
   if  Ci•••tl$_S11:8SIOIJ('valid _u••r')II
   (
      ecbo 'You are logged in a•: '.$  811:88I011(valid u•er•J.'   cbr />';
      ecbo •ca bref•"logout.pbp">Log  outc/a>cbr  />',-

   •|••
   (
      if Ci•••t($u•er1dll
      (
         // ai ban intentado conectar•• y no lo ban eoa•eguido
         echo 'Could not log you  in' 1
      }
      elae
      (
         // todavla no ban intentado coaectar.. o••  ban de•coaectado
         ecbo  'You are not logged in.cbr />':


      // proporcione un for1111lario para que•• conecten
```



```
      ecbo  'cfora ..tbod••po9t• actioa-•autbllain.pbp°>'1
      ecbo  •ctable>' 1       '
      ecbo  'ctr>ctd>O..rid:c/td>'1
      echo  •ctd>cinput type••taxt•  name••u•erid">c/td>c/tr>'1
      echo  'ctr>ctd>PU.-rd:c/td>'1
      ecbo  •ctd>cinput typ&•"p&••-rd"  name•"pa••-rd">c/td>c/tr>';
      ecbo  'ctr>ctd col8JIAD-"2" align••center°>'1
      ecbo  'cinput type••lnllllit• valua••Log in">c/td>c/tr>'1
      ecbo  c/teble>c/forl1>'1

7>
cbr>
ca br•f••...tier•_anly.pbp•..-..ber•  ••ctionc/a>
c/body>
c/btab
```

Esta secuencia de comandos incluye;por alguna razón, una compleja lógica, ya que muestra el formulario de inicio de sesión, también la acción para el mismo y contiene Hl'ML para intentos de inido de sesión satisfactorios y fallidos.

Las actividades del formulario dependen de la variable de sesión valid_uaer• La idea básica es que si un usuario se conecta satisfactoriamente, registraremosuna variable de sesión con el nombre$_SBSSIOII(•valid_user •J qüecontenp el Id. del usuario. Lo primero que hacemos en esta secuencia de conwndos es invocar session_start (). De esta formase carga la variabledesesiónvalid_usersise ha registrado.

En la primera pasada por la secuencia de comandos, no se aplica ninguna de las condiciones if y el usuario continúa hasta el final de la misma, donde le indicaremos que no se ha conectado y le o&eceremos un formulario para que lo haga:

```
echo •cfora —tbod•"p09t• action••autbmain.pbp">';
echo •cteble>' 1
ecbo •ctr>ctd>O••rid,c/td>';
ecbo •ctd>cinput typa••text• naae••u•erid">c/td>c/tr>'1
ecbo •ctr>ctd>Pa••-rd,c/td>'1
ecbo •ctd>cinput typa••p&•IIVOrd• na-•pa•8VOrd">c/td>c/tr>' i
ecbo •ctr>ctd col.p&n•"2" align••center°>':
ecbo 'cinput typ&•"•ubait• valu••"Log in">c/td>c/tr>'1
echo 'c/teble>c/fora>';
```

Cuando pulse el botónde envio del formulario, se vuelve a invocar esta secuencia de comandos y empezamos de nuevo desde el principio. En esta ocasión, k.-nemos un Id. de usuario y una contrasei\a que autenticar, almacenados como $_POST[•userid'] y$_POST['password'] .Siestasvariablesestmconftguradas, puamos al bloque de autenticad6n:

```
if  li•••tl$_f'.0ST('uaerid'))    .. ia•et(_POST('pa••-rd')))
(
   //del  uauario acaba de intentar conectar••
   $u..rid • $_POST(•u..rid')1
   $pa811VOrd .-. _P08T(pa811VOrd')1

   $db_conn • av-ql_comiect('localbo•t', •vebautb', •vebautb'11
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
if (11Y941li_coanec:c_•rrnoCII {
   ecbo 'coan•ccion co dac•IHI•• failn,•.my9qli_coaa.c:c_error(J1
   exic O,

SqiMry • '••l•cc •rfrOII au~h '
          .•vilera n-+1 $uNrid• •
          .• aDd pa•••paaa-rdC•Spa•—rd•I•;

Sneulc • se_coan-,.qu•ryl$q1Mryl:
```

Nos conectamos a una base de datos MySQL y comprobamos el Id. de usuario y la contrasei\a. Si hay una combinación que coincida en la base de datos, creamos la variable $_SESSION [•valid_uaer •J que contiene elld. de usuario de este usuario en concreto, por lo que a partir de ahora sabremós quién esti conectado.

```
if (Sn•ulc-,.1111a_rowe "'0 |
{
   // •l •e •ac:u.ncraD ea la IHI•• de dacoa, revi•cre el Id. de ueuario
   S_SISSIOll('valid_u••r'I • Su••rid;
}
SiDI_COGD-,.cloe•<I,
```

": Como ahora sabemos de quién se trata, no es necesario volver a mostrar el f~onulario de inicio de sesión. En su lugar, le indicamos que ya sabemos quién es y ~ ofrecemos la opción de desconectarse:

```
I• (ieHC cs__HSSIOII(•valid_u..r•111

ec:bo 'touar• logpd in aa, '.$_SBSSIOll('valid_ue•r~J..' cbr /:,,';
ec:b.o •ca hr•f•°logouC.php°,,Log oucc/a:,,cbr /,,,,
```

Si intentamos conectar al usuario y, por alguna razón, no lo conseguimos, tendremos. un Id. de usuario pero no una variable $_SESSION C•valid_uaer• J, por lo que podremos ofrecerle un mensaje de error:

```
if 11 ... c,sue•rid11
{
   // •i han iacuacado coaeccaru ~ro DO lo llan coa•evuido
   ecbo •could DOC log you in•:
```

Eao es todo para la secuencia de comandos principal. Seguidamente, analizaremos la p6gina de miembros, cuyo código mostramos en el listado 22.S.

UalNo ZZ.S. membera_only.php. *C6dlgo* para la seccl6n de mlembras de~ sitio Web para comp,ober la validez de loa ueuartos.

```
c'1plip
   -••ioa_•carc<I,

   ecb.o 'chl,.,....r• ODlyc/hl>'1
```



```
if 11.. ec(S_SISSIOlll('valid_u•er'111
{
   Kilo •cp:,,You are loggn iD - •.s_1a1:cm[•ftlid_1111•r•J.•c/p,.•,
   ecllo •cp.....a.r. ODly CODC—C ~ i.r.°c/p,,'¦

el••
{
   ecbo 'cp:,,!ou an DOC logged in.c/p:,,';
   echo •cp:,,Only logg•d in ...,.n •Y — chie pege.c/p:,,•,

echo 'ca href•°auclmain.php•:,,Back co MiA pagec/a:,,•;
```

Este código es muy sencillo. Todo lo que hace es iniciar una sesión y comprobar si la sesión actual contiene un usuario registrado. Pan ello comprueba si se ha definido el valor de $ SBSSION [•valid user •J. Si el usuario se ha conectado, le mostramos los contenidos de los miembíos; en caeo contrario le indicamos que no esti autorizado.

Por dltimo, la secuencia de comandos logout •php permite desconectar a un usuario del sistema. El código correspondiente se incluye en el listado 22.6.

U.a.do 22.1. logoulphp. Est8 secuencia de COlll811dol anula el registro de la variable de sesión y elimina la sesión.

```
c?php
   •e..loa_•carc()1

   // al•c.a. para probar •i •eecaban• coaeccadoe
   Sold uaer • $ S2SSION('valid ue•r'l1
   ua•et($ SBSSIOll(•valid ua•r')I;
   •e••ioa=d•ecroy(I; -
?:,,
chC•1:,,
cbody:,,
chl:,,Log OIICc/hl:,,
c?php
   if (leepcy(Sold_u•erl1
   {
      echo 'Loggn OUC.cbr /:,,';

   el••
   {
      // ai DO ••caban coaeccado• ~ro accedieron de algCID 80do auca ~gi-
      echo 'tou —r• noc loggn in, aad ao ba" aoc ~Hn logged ouc.cbr /,.,,

?:,,
cahnf••auclma:la.php•,.9acJt co —in ~e/a,,
c/body:,,
</hc11b
```

EL_HOLISTICO@HOTMAIL.COM - 2006

El código es muy lffflcillo pero lo analizaremos brevemente. lniciamo. una sesión, almacenamos el antiguo nombre de usuario del usuario, ~ulamos la variable valid_user y destruimos la sesión. Tras ello, mostramos al usuario un mensaje que ser4 düerente si se ha desconectado 0110 se ha conectado desde un principio.

Este sencillo conjunto de secuencias de comandos constituye la base de gran parte d~l trabajo que desarrollaremos en capítulos posteriores.

Lecturas adicionales

Puede encontrar m'8 información sobre las cookies en ht tp: / /hoae. netscape. com/newsref/std/cookie_spec.html.

A continuación

Prácticamente hemos terminado can esta parte del libro. Antes de pasar a los proyectos, .analizaremos brevemente algunos de los aspectos m'8 útiles de PHP que no hemos mencionado hasta el momento.

EL_HOLISTICO@HOTMAIL.COM - 2006

23

características útiles

Algunas de las funciones y características útiles de PHP no encajan en una categoría concreta, razón por la que las desaibiremos en este capítulo. Analizaremos loe siguientes aapectcs:

- Utilizar comillas mágicas
- Evaluar cadenas con eval ‹l
- Finalizar la ~jecución con die y exi t
- Serialización (variables y objetos)
- Obtener información sobre el entorno PHP
- Modificar temporalmente el entorno de ejecución
- Cargar extensiones PHP
- Resaltar código fuente
- Utilizar PHP en la línea de comandos

Utilizar comillas mágicas

Probablemente se haya fijado en que debe prestar especial atención al u~lizar el símbolo de comillas (‘ y •) y las barras invertidas (\) dentro de una cadena. PHP se puede confundir si utilizamos una instrucción de cadena como la siguiente:

```
echo ‘color‘ •tFFFFPF••;
```

y generará un error de arullisis. Para incluir comillas en una cadena, utilizaremos un tipo de comillas distinto al que se emplea para delimitar la cadena. Por ejemplo:

```
ecbo •color•  'IFFPPFP'";
```

o

```
ecbo 'color•  •trPFPn••,
```

son dos posibilidades válidas. El mismo problema se produce con las entradas de usuario, asf como con entradas o salidas de o hacia otros programas.

Si intentamos ejecutar una consulta como

```
in•ert into company valu•• ('Bob'• Auto Pert•'Il
```

se producirá una confusión similar en el analizador de MySQL.

Ya hemos visto el uso de addslaahes () y stripslashes () que escapan cualquier comilla simple, comilla doble, barra invertida y caracteres NULL.

PHP .dispone de una opción muy útil que ailade o elimina barras invertidas automática o mágicamente por nosotros. Con dos parámetros en nuestro archivo php • ini podemos activar o desactivarlas comillas mágicas para GET, POST, datos de cookies y otras fuentes. El valor de la directiva magic_quotes_gpc controla si se utilizan las comillas mágicas en operaciones GBT, POST y de cookies.

Al activar magic_qoutes_gpc, si alguien escribe "Bob\ Auto Parta" en un formulario de nuestro sitio, nuestra secuencia de comandos recibirá "Bob\ '• Auto Parta" porque a la a>milla se le aplica conversión de escape. Este comportamiento púede resultar muy 1ltil pero debe recordar cómo funciona para no olvidarse de eliminar las barras antes de devolver los datos al usuario. Esto es sencillo si el código se ejecuta en un servidor pero si tiene pensado distribuir su c6digo, puede que le interese que funcione con o sin a>millas mágicas. La función get_magic_quotea_gpc () devuelve 16 o, para indicamosel valor actual de aagic_quotes_gpc.

Esto iesulta de gran utilidad para probar si necesi-mos aplicar atripslashes ‹) a datos recibidos de un usuario. El valor de –gic_quotes_r-untime controla si las funciones que obtienen datos de bases de datos y de archivos utilizan las comillas mágicas. Para obtener el valor de —gic_quote_runtime, utilice la función get_magic_quotes_runtime (), ~ devuelve 1 6 0. Las comillas mágicas se pueden desactivar en una secuencia de comandos concreta por medio de la función aet_magic_quotea_runtime().

Evaluar cadenas: eval()

La función eval () evalúa cadenas como si se tratara de código PHP. Por ejemplo:

```
aval ( •ecbo 'Hallo Norlil•;• 1;
```


adopta los contenidos de la cadena y los ejecuta. Esta línea genera el mismo nsultado que

```
echo 'hllo  llorld •,
```

En determinados casos eval () puede resaltar muy dtil. Puede que le interese almacenar bloques de c6digo en una bue de dalm y recuperarlos y aplicarle eval () posterionnente. Puede que quiera generar código en un bucle y, tras ello, utilizar eval () para ejecutarlo. La aplicación más habitual de eval () es como parte de un sistema de plantillas. Puede cargar una mezcla de HTML. PHP y texto sin procesar de una base de datos. Su sistema de plantillas puede aplicar formato a este contenido y, tras ello, ejecutarlo a travm de eval ‹› para ejecutar cualquier c6digo de PHP.

Puede tambWn utilizar eval () para ac:tualizar o corregir código existente. Si dispone de gran cantidad de secuencia de comandos en las que es necesario aplicar un cambio predecible, podría (aunque no seria muy eficaz) escribir una secuencia de coawndos que cargue una secuencia de comandos antigua en una cadena, ejecule regexp para realizar los cambios y tru ello utilice eval () para ejecutar la secuencia de comandos modificada.

Incluso podemos pensar que haya alguien de mucha confianza en alguna parte que quiera permitir la entrada y ejecuci6n de código PHP en su navegador.

Finalizar la ejecución: die y exit

Hasta ahora hemos utilizado la instrucción exi t para detener la ejecudón de una secuencia de comandos. Como recordará, aparece en su propia línea, a>mo mostramos a continuadón:

```
exit1
```

No devuelve nada. También podemos utilizar su alias die ( ) .

Si necesitamos una finalización más útil, podemos pasar un parámetroa exit (), que podemos utilizar para mostrar un mensaje de error o para ejecutar una funci6n antes de finalizar una secuencia de comandos. Seguramente le resulte familiar a los programadores de Peri. Por ejemplo:

```
exit('Script  ending now'I;
```

Normalmente se utiliza or con una instruccidn que puede fallar, como por ejemplo al abrir un archivo o conectarse a una base de datos:

```
my•ql_queryl$query) or die(•could not execute query•Il
```

En lugar de imprimir simplemente un mensaje de error, puede invocar una última función antes de que finalice la secuencia de comandos:

```
:unctian err_ug()
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
ret11r11 •MySQL error •••'  '.ayaql_errorCI;

myaql_query(Squery1  or die(err_ma9(lll
```

Puede resultar muy útil para indicarle al usuario la razón por la que ha fallado la secuencia de comandos, para cerrar elementos HTML o para eliminar una página medio completada del búfer de salida.

También podriamos enviamos un correo electrónico a nosotros mismos para saber si se ha producido un error grave o ailadir errores a un archivo de registro.

Serialización

La serialización es el proceso de convertir todo lo que se pueda almacenar en una variable PHP o en un objeto en un flujo de bytes para almacenarlo en una base de datos o pasarlo a través de un URL de una página a otra. Sin esto, resultaría muy c~mplicado almacenar o pasar todos los contenidos de una matriz o de un objeto.

Desde la aparición del control de sesiones, su utilidad se ha visto reducida. La señalización de datos se aplica principalmente al tipo de cosas para las que ahora utilizamos el control de sesión. De hecho, las funciones de control de sesión serializan v~ables de sesión para almacenarlas entre solicitudes PHP. Sin embargo, puede q• le interese almacenar un objeto o matriz PHP en un archivo o en un documento. Si es su intención, hay dos funciones que debe conocer: serialize (l y -u1111erialize e>. Puede invocar la Jüñaon sirlaHzilTae estiToniia:

```
S••rial_~ject • Hrialize($ay_objectl
```

Si quiere saber lo que realmente hace la serialización, basta con fijarse en lo que devuelve la función. Convierte los contenidos de un objeto o de una matriz en una cadena. Por ejemplo, fijémonos en el resultado de la ejecución de serial i ze (l en un sencillo objeto de empleado, definido e instanciado de esta forma:

```
el••• 911Ployee
(
   var $naae1
   var $employee_id;


Stbia_emp • new •11Ploy••1
Stbia_emp->na.. • 'rred';
Stbi•=•mp·>e.ployee_id  • 5324;
```

Si lo serializamos y lo duplicamos en un navegador, el resultado sen

```
0,1, •employee•:2 :"{a:4, · —— · ;a,4: •rred" ;a:11: •employ-_id•; i153241}
```

Es butante sencillo ver la reladón entre los datos del objeto original y los datos serializa4os.

Como los datos serializados son sólo texto, podemos escribirlos en una base de datos o donde queramos. Sepa que es necesario aplicar addda•hH (1 a cualquier dato antes de escribirlo en una base de datos, como de costumbre. Aprecian la necesidad de esto si se fija en las comillas de la cadena serializada anteriormente.

Para recuperar el objeto, se invoca unserialize (1:

```
Snew_~ject • unaerialize(Saerial_~jectl;
```

Evidentemente, si ha invocado add81Hhea (1 antes de incluir el objeto en la base de datos, tendra que invocar atrip•lHhea (l antes de deserializar la cadena. Otro aspecto que conviene destacar al serializar clues o al utilizarlas como variables de sesión es que PHP necesita conocer la estructura de una clase antes de poder volver a instanciarla. Por ello, será necesario incluir el archivo de definición de clase antes de invocar aeaa ion_atart () o unserialize ().

Obtener información sobre el entorno de PHP

Existen distintas funciones que podemos utilizar para encontrar información sobre la configuración de PHP.

Saber qué extensiones se han cargado

Es muy sencmo saber qué conjuntos de hmciones est4n dispoñible°$y-qué funciones estút disponibles en cada uno de dichos conjuntos. Basta con utilizar las funciones get_loaded_extenaions () y get_exten•ion_func• n.

La función get_loaded_extenaions O devuelve una matriz de todos los conjuntos de funciones disponibles actualmente para PHP. Por su parte, si proporclonamos el nombre de una determinado conjunto de funciones o una extensión, get_extensiona_funca ‹l devuelve una matriz de las fundones de dicho conjunto.

El código del listado 23.1 enumera todas las funciones disponibles para su instalación PHP al utilizar estas dos funciones.

Ultado 23.1. list_functlons.php. Esta sec:uencia de comandos enumena todas las extensiones disponibles para PHP y en cada extensi6n lncluye una lista de las funciones de dicha extensl6n.

```
c?pbp
   ecbo 'conjllltoa de funcione• adllitido• en eata inatalaci6n,cbr />'i
   $estenaiona • 9et_loadad_extenaiona(I;
   foteach ($estenaion• •• seacb_extl
   {
     ecbo "$eacb est cbr />"1
     ecbo 'cul>,7
     $ext_fllllca • get_estenaion_fllllce(Seach_extl;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
foreach($ext_funca •• $f1111cl
{
   echo •cli» Sfunc c/li»•1

echo 'c/lll»'1

?»
```

Ffjese en que la fundón get_loaded_extensions (1 no adopta ningún paúmetro y que la función get_extension_func• () adopta como único paúmetro el nombre de la extensión.

Esta información puede resultar muy lltil para saber si hemos instalado satisfactoriamente una extensión o si trata de escribir código portitil que genere lltiles mensajes de diagn6atico al inltalarae.

Identificar al propietario de la secuencia de comandos

Podqtoa descubrir el usuario propietario de la secuencia de comandos en ejecud6n con una llamada a la función get_current_user(), como se muestra a continuad6n:

```
acilo get_current_uaarll1
```

En ocasiones puede resultar muy útil para resolver problemas relacionados con loa permisos.

Saber cuándo se ha modificado una secuencia de comandos

Una operadón muy habitual consiste en ai\adir una llltima fecha de modificad6n a cada una de las pAginas de un sitio.

Para comprobar la llltima fecha de modificadón de una secuencia de comandos basta con utilizar la funcióngetlastmod (1 (el nombre de la fuñd6n no incluye un guión bajo) de esta forma:

```
echo date(•g:i  a, j N Y\,getla•tllOdCll1
```

La función devuelve una marca de tiempo de Unix que podemos aftadir a date () como hemos hecho en el ejemplo para generar una fecha legible.

Carga dinámica de extensiones

Se pueden cargar bibliotecas de extensiones dinAmicas durante el tiempo de ejecución, si no estAn compiladas, por medio de la fund6n dl (), Esta función espera como parámetro el nombre del archivo que contiene la biblioteca. En Unix,

se trata de nombres de archivo que tenninan en •so; en W'mdows, terminan en . dl l. Un ejemplo de una invocación de dl () seria el siguiente:

```
dl(•php_ftp.dll'I;
```

De esta forma se carga dinAmicamente la extensión FTP (en un equipo ba¡o Windows).

No es necesario especificar el directorio en el que se encuentra el archivo; basta conconfigurarloenelarchivophp.ini.Unadirectivadenominadaextenaion_dir especificar, el directorio en el que PHP buscar6 bibliotecas para cargarlas din6micamente.

Si tiene problemas para cargar extemicnes dinAmicamente, debe comprobar si en su ardüvo php. ini se incluye la directiva enable_dl. Si estt desactivada, no podú cargar extensiones dinúnicamente. Puede que si el ordenador con el que trabaja no es suyo, estf desactivada por motivos de seguridad. No podr6 utilizar dl () si PHP se ejecuta en modo seguro.

Modificar temporalmente el entorno de ejecución

Puede ver las directivas configuradas en el archivo php . ini otlmbiarlas mientras dure una determinada secuencia de comandos, lo que por ejélilplo puede resultar muy lltil junto con la directiva .. x_execution_time si •bemos que la secuencia de comandos tardará en ejecutarse.

Podemos acceder y modificar las directivas por medio de las funciones ini_get () e ini_set (). En el listado 23.2 se incluye una sencilla secuencia de comandos que utiliza estas funciones.

LIBtNo 23.2. inter.php. Sec:uencill de comancloa que restableceYMBblea del an:Nvo php.lnl.

```
c?php
   $old_ux_execution_ti• • ini__Ht(•ux_execution_ti__',  12011
   echo •old ti.eout ia Sold_ux_execution_ti•e cbr /»•1

   $max_execution_time • ini_get(•max_execution_ti..'11
   echo •new timeout ia $MX_execution_ti.. cbr /»•1
```

La fundón ini_set () adopta dos par6metros. El primero es el nombre de la directiva de configuradón de php . ini que queremos cambiar y el segundo, el valor por el que la cambiaremos. Devuelve el valor anterior de la directiva.

En este caso, restablecemos el valor de durad6n múimo predeterminado de una secuencia de comandos, 30 segundos, para que se ejecute durante 120 segundos.

La función ini_get (1 simplemente comprueba el valor de una determinada directiva de configuración. El nombre de la directiva se debe pasar a la misma en

EL_HOLISTICO@HOTMAIL.COM - 2006

forma de cadena. En este caso sólo lo utilizamos para comprobar que el valor ha cambiado realmente.

No todas las opciones lNI se pueden configurar de esta forma. Cada opción cuenta con un nivel en el que se puede configurar, niveles que detallamos a continuación:

- PHP INI USER: Puede cambiar estos valores en sus secuencias de comandos por i'iii_set ().
- PHP INI PERDIR: Puede cambiar estos valores en los archivos php. ini, .hticceis o httpd. conf si utiliza Apache. La posibilidad de cambiarlos en archivo .htacce•• significa que podemos modificar estos valores en cada directorio, de ahf el nombre.
- PHP INI SYSTEM: Puede cambiar estos valores en los archivos php • ini o httpd.c;;-nf.
- PHP INI ALL: Puede cambiar estos valores de cualquiera de las formas anteriores: es decir, en una secuencia de comandos, en un archivo • htacceH o en SU$ archivos php. ini o httpd. conf.

El conjunto completo de opciones ini y los niveles en que se pueden configurar se recogen en el manual de PHP (http://www. php. net/ ini_set).

Resaltar código fuente

PHP incorpora un resaltador de sintaxis similar a muchos IDE. En concreto, resulta muy útil para compartir código con terceros o presentarlo en una ptgina Web para su análisis.

Las funciones show_source () y highlight_file () son idénticas (de hecho la primera es un alias de la segunda). Ambas funciones aceptan como parámetro un nombre de archivo (archivo que debe ser un archivo PHP ya que, en caso contrario, el resultado que obtendremos no tendrá mucho sentido). Por ejemplo,

```
•Arcstinqrrs/ ... Operqrcc
```

El archivo se duplicará en el navegador con un texto resaltado en varios colores en función de si se trata de un comentario, una cadena, una palabra clave o HTML El resultado se imprime sobre un color de fondo. El contenido que no encaja en ninguna de estas categorfas se imprime en el color predeterminado.

La función highlight _string {) funciona del mismo modo pero adopta como parúnetro una cadena que imprime en el navegador con un formato de sintaxis resaltado. Podemos definir los colores para el resalte de la sintaxis en el archivo php. ini, en la sección que mostramos a continuación:

```
, eoior.  for Syntaa  Hlghligbtiag  mode
bigblipt.atring      •  IDDOOOO
highlight.⊂∼nt       •  IPPIO00
higbligbt.keyvord       I007700
highlight.bg         - - - - - - - - .... rr
highlight.default    •  IO000II
highlight.htal       •  IOOOOOO
```

Los colores se muestran en formato RGB HTML estándar.

Utilizar PHP en la línea de comandos

Puede escribir o descargar muchos pequei\os programas y ejecutarlos en la linea de comandos. Si se trata de un sistema Unix, estos programas se suelen escribir en un lenguaje de secuencias de comandos de núcleo o en Peri. Si se trata de un sistema de Windows, se escriben como lenguaje por lotes.

Probablemente la primera vez que utilice PHP será para un proyecto Web, pero las mismas prestaciones de procesamiento de texto que lo convierten en un potente lenguaje Web lo convierten en una potente utilidad de linea de comandos.

Hay tres formas de ejecutar una secuencia de comandos de PHP en la linea- de comandos: desde un archivo, a travé de una canalización o directamente desde la linea de comandos. Para ejecutar una secuencia de comandos en un archivo, compruebe que el ejecutable de PHP (php o php . exe, en función de su sistema operativo) se encuentra en su ruta e invóquelo con el nombre de la secuencia de comandos como argumento. Veamos un ejemplo:

```
plip ay•cript.php
```

El archivo myscript. php es un archivo normal de PHP, por lo que ∼luye.la sintaxis habitual de PHP dentro de etiquetas de PHP.

Para pasar código a travé de una canalización, puede ejecutar cualquier programa que genere como resultado una secuencia de comandos de PHP válida y canalizarla hasta el ejecutable php. El siguiente ejemplo utiliza el programa echo para generar un programa de una sola línea:

```
echo  'c?php  for($i•l;  $ic10¡  Si++l  echo  Si;  ?>'  |  pbp
```

Como en el caso anterior, el código de PHP se encierra entre etiquetas de PHP (<?php y ?>). Fijese también en que se trata del programa de línea de comandos echo, no de la construcción del lenguaje del mismo nombre.

Resultaría más sencillo pasar un programa de una linea como &te directamente desde la linea de comandos, como vemos en el siguiente ejemplo:

```
pbp  -r  'for($i•l;  $1<10l  $1++l  echo  $1;  '
```

Es una situación un tanto düerente. El código de PHP pasado en esta cadena no se encierra entre etiquetas de PHP, ya que si lo hacemos se genera un error sintáctico. Los programas PHP que puede escribir para su utilización en la linea de comandos son innumerables. Puede diseftar instaladores para sus aplicaciones de PHP, crear

EL_HOLISTICO@HOTMAIL.COM - 2006

una r6plda aecuenda ele comandos para aplicar fonnalv a un archivo de lulo Ulles de importarlo a su bue de datoe, incluso realizar tueu repetitlYal que l8tga que ejecutar clelcle la linea de colMlldc»; un buen candidato Nda una NCUelda ele comandoll para copiar lOdos IUS an:hivos de PHP, ... y estructura ele tablas MySQL delde un aerridor Web de dlNfto hMta 011o de pn,ducd6n.

A continuación

In la liguienle pule wlizaremae una aerie de pn,yedDI pr6c:ti,cal ieiatiYUMn-1e complicados que utilizan PHP y MySQL Esloe proyectm CIDIIStituyen ejemploe muy 11tiles para tueu aimilanl que l8tga que clesurollar y demueltran el \IIO ele PHP y MySQL en proyectoe ele mayor tamafto.

In el siguiente capftulo IIOII centruemoe en algunoa problemal a los que noe enfnntaremae al codificar graneles proyeclm con PHP. Abordaremol princ:ipiae de ingesµerfa de lOftwue como por ejemplo el dudo, la docwMntad6n y la psti6n decánbi6e.

EL_HOLISTICO@HOTMAIL.COM - 2006

Parte V

Crear proyectos PHPyMySQL prac treos

24

u~t1~1zar PHP y 1\AySQL en grandes proyectos

En las anteriores partes de ~te libro hemos analizado distintos componentes y usos de PHP y MySQL. Aunque hemos intentado que todos los ejemplos fueran interesantes y relevantes, todos han sido muy sencillos, formados por una o dos secuencias de comandos de hasta 100 líneas de código aproximadamente.

Al generar aplicaciones Web reales, las cosas no suelen ser tan sencillas, Hace tiempo, se consideraba que un sitio Web era interactivo cuando éste incluía correo de formulario y poco más. Sin embargo, en la actualidad, los sitios Web se han convertido en aplicaciones Web, es decir, un producto de software convencional que se entrega por la Web. Este cambio de enfoque significa un cambio de escala. Los sitios Web pasan de tener un puñado de secuencias de comandos a miles y miles de líneas de código. Los proyectos de este tamaño requieren la misma planificad6n y gesti6n que cualquier otro desarrollo c;le software.

Antes de pasar a analizar los proyectos en este apartado del libro, nos detendremos en algunas de las técnicas que podemos utilizar para gestionar proyectos Web de gran tamaño. Se trata de un arte en expansi6n y resulta muy complicado dominarlo correctamente, lo que comprobará si analiza el mercaqo.

En este capitulo nos detendremos en los siguientes aspe¿tos:

- Aplicar ingeniería de software al desarrollo Web
- Planificar y ejecutar un proyecto de aplicad6n Web
- Reutilizar c6digo
- Escribir c6digo mantenible

EL_HOLISTICO@HOTMAIL.COM - 2006

- Implementar el control de versiones
- Seleccionar un enlomo de desarrollo
- Documentar un proyecto
- Prototip~
- Separar lógica, contenido y presentadón: PHP, HTML y CSS
- Optimizar el código

Aplicar ingeniería de software al desarrollo Web

Como seguramente ya sabrá, la ingeniería de software es la aplicación de un enfoque sistemático y cuantificable al desarrollo de software. Es decir, se trata de la aplicadón de los prindpios de ingeniería al diseño de software. También es un enfoque q_ue falta en muchos proyectos Web, prindpalmente debido a dos razones.

La primera es que el desarrollo Web se suele gestionar de la misma forma que el desarrollo de informes escritos. Es un ejerddo de estructura de documentos, disefto gráfico y producción, un paradigma orientado a documentos. Es totalmente válido para sitios de pequetlo o mediano tamafto pero al aumentar el número de contenidos dinámicos de los sitios Web hasta alcanzar el nivel en el que dichos sitios of,ecen servicios en lugar de documentos, este paradigma ya no sirve. Hay mucha gente que ni siquiera piensa en utilizar prácticas de ingeniería de software en un proyecto Web.

La segunda razón por la que no se ha utilizado la ingeniería de software es que el desarrollo de aplicadones Web es diferente al de aplicaciones normales. Los plazos son mucho más reducidos y la presión por finalizar el sitio es constante. La ingeniería de software consiste en hacer cosas de forma ordenada y planificada así como en dedicar tiempo a la plariificadón. Con los proyectos Web, normalmente se tiene la sensadón de que no hay tiempo para planificar. ...

Si no planificamos un proyecto Web, sufriremos los mismos problemas que si no planificamos cualquier proyecto de software: aplicadones defectuosas, fechas de entrega sin cumplir y código ilegible. Por ello, el truco consiste en identificar las partes de la ingeniería de software que funcionan en esta nueva disciplina del desarrollo de aplicadones Web y en rechazar las que no lo hacen.

Planificar y ejecutar un proyecto de aplicación Web

No existe una metodología o ciclo vital idóneo para un proyecto Web. No obstante, existen distintos aspee~ que se pueden tener en cuenta a la hora de preparar

un proyecto. Los enumeramos ~os y, en los siguientes apartados, describiremos algunos con mayor detalle. Estas consideraciones tienen un orden concreto pero no es necesario que lo siga si no se ajusta a su proyecto. El élfuis recae en estar alerta sobre los poaibles errores y en seleccionar las limas que nos puedan servir.

- Antes de empezar, piense en lo que quiere generar. Piense en el objetivo final. Pienae en quién va a utilizar su aplicación Web, es dedr, en su público de destino. Muchos proyectos Web ticnicamente perfedos no funcionan porque nadie ha comprobado si hay usuarios interesados en dicha aplicación.
- Pruebe su aplicad6n y divfdala en componentes. ¿De quf partes o pasos se compone su aplicación? ¿Cómo funciona cada uno de estos componentes? ¿Cómo se combinan entre sí? El disetlo de escenarios, guiones gráficos o incluso de casos prácticos puede ser muy útil para determinar estae aspectos.
- Una vez confeccionada la lista de componentes, compruebe cuáles existen ya. Si un módulo preconfigurado ya cuenta con dicha fundonalidad, utilícelo. No olvide buscar código existente dentro y fuera de su organizadón. Existen muchos componentes de código gratuitos, en especial en la comunidad de código abierto. Determine quf partes del código debe escribir desde cero y cuánto le llevará hacerlo.
- Tome decisiones sobre aspectos del proceso, algo que se suele ignor:ar en los proyectos Web. Por aspectos del proceso nos ~f~ a a~ ~mo los estándares de codificación, estructuras de directorios, admJDJStraaón del control de versiones, entorno de desarrollo, nivel y estándares de documentación, y asignaá6n de tareas a los integrantes del equipo.
- Genere un prototipo, basado en toda la información anterior. Mufstrelo a los usuarios. Repftalo.
- Recuerde que, en todo esto, es muy importante separar el contenido y la lógica de la aplicación, como veremos posteriormente con mayor detalle.
- Realice todas las optimizaciones que considere oportunas.
- Mientras avance, pruebe concienzudamente la aplicadón, como haría con cualquier otro proyecto de desarrollo de software.

Reutilizar código

Los programadores suelen cometer el error de volv!r a escribir ~igo.que ya existe. Una vez identificados los componentes necesanos para su aplicaaón o, a menor escala, las funáones que necesita, compruebe lo que está disponible antes de iniciar el desarrollo.

Uno de los puntos fuertes de PHP como lenguaje es. la ~tensa bibli~eca de funciones que incorpora. Siempre debería comprobar si existe una función que

EL_HOLISTICO@HOTMAIL.COM - 2006

haga lo que quiere conseguir. No le resultará difícil encontrar la que busca. Una buena forma de hacerlo consiste en examinar el manual por grupos de funciones.

En ocasiones los programadores vuelven a escribir funciones accidentalmente ya que no han comprobado si una de las funciones existentes en el manual ofrece la funcionalidad que necesitan. Debería marcar siempre el manual si se encuentra en lfnea o descargar la venión actual y examinarla localmente. Sepa, sin embargo, que el manual en lfllea se actualiza frecuentemente y que tambiln tiene la ventaja de poder consultar el manual anotado. tste es una magnifica fuente que contiene comentarios, sugerencias y ejemplos de código de otros 1,15uarios que suelen resolver las dudas que pueda tener después de leer el manual ~sico. La versión en español del manual se encuentra en la página http://www.php.net/manual/

••· Algunos programadores con experiencia en un lenguaje diferente pueden verse tentadoe a escribir funciones contenedoras para cambiar el nombre de las funciones PHP de manera que coincidan con las del lenguaje que conocen. Esta práctica no es nada aconsejable, ya que dificulta la lectura y el mantenimiento del código para otros. Si está aprendiendo un nuevo lenguaje, tendrá que aprender a utilizarlo correctamente. Adem6s, la inclusión de un nivel de llamada de funciones de esta manera ralentizaría el código. Por todas estas razones, es un enfoque que debería evitar. Si la funcionalidad que necesita no se encuentra en la biblioteca PHP principal, tiene dos.opciones. Si necesita algo muy sencillo, puede optar por escribir su 'Pr$>pia ~citan u objeto. Sin embargo, si pretende generar una funcionalidad m6s cómpl—, coa,o un carro de la compra, un sistema Web de correo electrónico o foros Web, nQ Se sorprenda si alguien ya lo ha generado. Una de las ventajas de ttaba;ar;iñ 14 ~omi.ii'iidad de c:6d:igo abierto es que et cddlgo para esa- ttpo de compoaintes de aplicación está disponible gratuitamente. Si ~tra ?' componente-similar al que necesita, aunque no sea exacto; puede utilizar eJ.código fuente como punto de partida y modificar o generar el suyo propio.

Si opta por diaeñar sus propias funciones o componentes, debería ofrecerlas a la comunidad PHP en cuanto termine. Es el principio que convierte a la comunidad de programadores PHP en algo tan útil y activo.

Escribir código que perdure

El problema del mantenimiento es in&avalorado en las aplicaciones Web, en especial por que las escribimos a toda prisa. Empezar el código y acabarlo rápidamente suele ser m6s importante que planificarlo con antelación. Sin embargo, con una pequeña revisión previa nos ahorraremos gran parte del camino cuando sea necesario generar la siguiente iteración de la aplicación.

Estándares de codificación

La mayoría de las organizaciones rr disponen de estÚldares de codificación, directrices de estilo para seleccionar nombres de archivos y variables, dirl!ctric:es

para comentar código, directrices para identificarlo, etc. Debido al paradigma de código que aplicamos previamente al desarrollo Web, los estándares de codificación se han infravalorado. Si el código lo disefta personalmenteo forma parte de un pequeño equipo, puede que no le di la importancia que tiene. No lo haga por temor a que el equipo o el proyecto puedan crecer. No sólo obtendrá un producto sin sentido, sino tambifn un montón de programadores que no saben ni qué hacer con el código existente.

Definir convenciones de nomenclatura

Los objetivos de la definición de una convención de nomenclatura son:

- Facilitar la lectura del código. Si define variables y nombres de funciones con sentido, podrá leer el código prácticamente como si se tratara de una frase normal o, al menos, pseudocódigo.
- Que los nombres de identificadores sean fáciles de recordar. Si sus identificadores tienen un formato coherente, resultar6 más sencillo recordar el nombre de una determinada función o variable.

Los nombres de variables deberían describir los datos que contienen. Si se trata de almacenar el apellido de alguien, llame a la variable $apellido. Tendrá que encontrar un equilibrio entre longitud y legibilidad. Por ejemplo, si almacena el nombre en $n, será m6s sencillo de escribir pero el código será más diffdl de entender. Si al~ac~ el noml:>~ en ~~~lli~~¿~L~!~ª~!~—~~;~a!~más claro, pero tardará más en escribirlo (y es más proclive a errores ortográficos) y realmente no ayuda tanto.

Tendrá que tomar una decisión en cuanto a las mayúsculas. En PHP, los nombres de variables distinguen entre mayúsculas y minúsculas, como hemos mencionado anteriormente. Tendr6 que decidir si los nombres de las variables van en minúscula, en mayúscula o una mezcla de ambas, por ejemplo, las primeras letru de las palabras en mayúscula. Por lo general se utilizan siempre minúsculas, ya que resulta más sencillo de recordar.

También es aconsejable distinguir entre variables y constantes con mayúsculas. Una práctica habitual consiste en utilizar minúsculas con las variables (por ejemplo, $result) y mayúsculas con las constantes (por ejemplo, PI).

Una práctica desaconsejable que algunos programadores emplean es utilizar dos variables con el mismo nombre pero en minúsculas y mayúsculas, como por ejemplo $name y $Hame. Espero que sepa por qué no es aconsejable.

Tambim conviene evitar modelos complicadoe, como por ejemplo $1faReZ, ya que nadie podrá recordar cómo funciona.

Debería pensar en qué esquema utilizará con nombres de variables de varias palabras. Por ejemplo, ya hemos visto los siguientes esquemas: .•

$uaerna-
$uaer_na_
$U••r.._

EL_HOLISTICO@HOTMAIL.COM - 2006

Puede utilizar cualquiera, pero higalo siempre de forma coherente. Tambifn puede establecer un límite máximo de dos a tres palabras en un nombre de variable.

Los nombres de funciones tienen la llÚSJlla consideración, con algunos aspectos adicionales. Los nombres de funciones suelen-estar orientados a verbos. Por ejemplo, funciones de PHP como addslaahes () o uyql_connect () describen lo que van a hacer o los par6metros con los que se pasan. Esto mejora considerablemente la legibilidad del código. Debe saber que estas dos funciones tienen un esquema de nomenclatura diferente para utilizar nombres de funciones de varias palabras. A este respecto, las funciones de PHP no son coherentes, probablem_ente por la gran cantidad de gente que las ha escrito, pero principalmente porque muchos nombres de funciones se han adoptado sin cambiar de distintos lenguajes y API. También debe recordar que los nombres de función no distinguen entre mayúculas y minúsculas en PHP. Probablemente debería adoptar un formato concreto, para evitar confusiones.

Puede optar por emplear el esquema de nombres de módulo utilizado en muchos módulos de PHP, es decir, se aftade el nombre de la función como prefijo al módulo PHP. Por ejemplo, todas las funciones MySQL empiezan por mysqli_ y todas las funciones IMAP con imap_. Si por ejemplo tiene un módulo de carro de la compra en su código, podría aftadir el prefijo carro_ a la función de dicho módulo.

No obstante, como PHPS proporciona tanto una interfaz basada en procedimientos como una interfaz orientada a objetos, los nombres de las funciones son dilérentes. Por lo general, las funciones basadas en procedimientos utilizan guiones bajos (my_function) y las orientadas a objetos utilizan el formato myPunction () .

'in definitiva, no importa qué convenciones y est4ndares utilice cuando escriba el código, siempre que aplique directrices coherentes.

Comentar el código

Todos los programas deberían incluir un cierto número de comentarios. Y se preguntará cuál este-número. Normalmente debería aftadir comentarios a _cada uno de los siguientes elementos:

- Archivos, ya san aec:uenclM de comando completas o uddvoa illduidos: Cada archivo debe tener un comentario que indique lo que es el archivo, para qué es, quifn lo ha escrito y cuándo se ha actualizado.
- Pundonell: Los comentarios de una función deben especificar qué hace la función, qué entrada espera y qué devuelve.
- Claee: Los comentarios deben describir el objetivo de la clase. Los métodos de clases deben tener el mismo tipo y nivel de comentarios que cualquier otra función.
- Pragmentoa de c6dlgo dmtro de una secuencia de c:mnandoa o ele una fund6n: Es muy útil escribir una secuencia de comandos que comience con una serie de comentarios y tras ello completar el código de cada sección. Una secuencia de comandos inicial seria como ésta:


```
c7
// valida dato• introducido•
// anvta a la baH da datoa
// informe da loa ren1tadoa
'''
```

Resulta muy útil por que una vez completadas todas las secciones con llamadas de funciones u otros elementm, el ~igo ya está ~mentado.

- C6cligo CClll\plejo: Cuando le lleve mucho tiempo escn"biralgo o lo tenga que hacer de forma poco habitual, escn"ba un comentario que explique por qué lo ha hecho. De esta forma, la próxima vez que lea el código sabrá exactamente lo que tiene que hacer.

Otra norma general que debe seguir es comentar el código mientras avanza. Puede pensar en volver atrás y comentar el código una vez terminado el proyecto. Le garantizo que no sucederá, a menos que tenga unas fechas de entrega menos estrictas y mayor autodisciplina que nosotros.

Sangrado

Al igual que en cualquier lenguaje de programación, es aconsejable sangrar el código de forma coherente y con sentido. Es igual que redactar un resumen o una carta de negocios. El sangrado facilita la lectura del código y hace que se entienda con mayor rapidez.

Por lo general, cualquier bloque de programación dentro de una estructura de control debería sangrarse del código adyacente. El grado de sangrado debe ser perceptible (es decir, más de un espado) pero no excesivo. Personalmente creo que no debería utilizar tabulaciones.

Aunque son más fáciles de aftadir, consumen gran cantidad de espacio en la pantalla de la mayoría de los usuarios. Suelo utilizar un nivel de sangrado de dos a tres espacios en todos los proyectm.

La forma de disponer las llaves tambifn es un problema. Las dos lKnicas más habituales son las siguientes:

Tbica 1:

```
if (condici6lll (
   // bap algo
```

Tbica 2:

```
if (condici6nl
(
   // baga algo úa
```

Puede utilizar cualquiera de las dos, pero la que seleccione debe utilizarla con coherencia a lo largo del proyecto para evitar confusiones.

EL_HOLISTICO@HOTMAIL.COM - 2006

Dividir el código

El código monolftico de gran tamaño es horrible. Mucha gente incluye en una línea principal de código una enorme secuencia de comandos que se encarga de todo. Es mucho mejor dividir el código en funciones y/ o clases y añadir los elementos relacionados en archivos. Por ejemplo puede incluir todas las funciones relacionadas con bases de datos en un archivo con el nombre dbfunctions. php.

Entre las razones para dividir el código en fragmentos de menor tamaño podemos destacar Ju siguientes:

- Resulta mú sencillo leer y comprender el código.
- El código es más reutilizable y se minimizan las redundancias. Por ejemplo, el archivo dbfunctions. php anterior se podría reutilizar en todas las secuencias de comandos que quiera conectar a la base de datos. Si tuviera que cambiar su funcionamiento, sólo tendría que hacerlo en un punto.
- Facilita el trabajo en equipo. Si el código se divide en componentes, puede asignar la responsabilidad de dichos componentes a distintos integrantes del equipo. También evitará que un programador tenga que esperar a que otro termine con un archivo para proseguir con su trabajo.

Al iniciar un proyecto, deberla pensar en cómo va a dividirlo en componentes. Estóiimplica establecer líneas entre áreas de funcionalidad, pero no se preocupe ya que puede cambiar después dtt~menzarun~yecto. También debe decidir qué componentes hay que generar en primer lugar, qué componentes dependen de otros.y la fecha lfmite para desarrollarlos.

.. Incluso si todós los integrantes dél equipo trabajan con todos los fragmentos de código, es aconsejable asignar la responsabilidad principal de cada componente a una persona en conaeto. De esta forma, habrá un responsable por si algo sale mal con el componente. También debería haber alguien que actúe como director, es decir, una persona que compruebe que todos los componentes están preparados y se combinan con los componentes restantes. Esta persona también se encarga del control de versión, que analizaremos más adelante. Puede tratarse del director del proyecto o se puede asignar como responsabilidad independiente.

Utilizar una estructura de directorios estándar

Al iniciar un proyecto, es necesario pensar en cómo se reflejará la estructura de los componentes en la estructura de directorios del sitio Web. Al igual que no es aconsejable utilizar una única secuencia de comandos de gran tamaño que contenga toda la funcionalidad, tampoco es aconsejable tener un directorio gigante que contenga todo. Decida cómo va a dividirlo entre componentes, lógica, contenido y bibliotecas de código compartido. Documente su estructura y asegúrese de que todo el que trabaje en el proyecto dispone de una copia, lo que nos lleva al siguiente punto.

Documentar y compartir funciones internas

Al desarrollar bibliotecas de funciones es aconsejable que estén disponibles para el resto de programadores del equipo. Habitualmente, cada uno de los programadores de un equipo escribe su propio conjunto de funciones de base de datos, de fecha o de depuración, lo que supone una pérdida de tiempo. Es mejor compartir las funciones y las clases. Recuerde que incluso si el código se almacena en una zona o en un directorio disponible para todo el equipo, no lo sabrán a menos que se lo indique. Debe desarrollar un sistema para documentar bibliotecas de funciones internas y para compartirlas con los programadores de su equipo.

Implementar el control de versiones

El control de versiones es el arte de la gestión de cambios simultáneos aplicada al desarrollo de software. Los sistemas de control de versiones funcionan como un archivo central y proporcionan una interfaz controlada para acceder y compartir el código (y también la documentación).

Imagine que tiene que mejorar un código pero, por accidente, lo divide y no puede recuperar la forma que tenía, haga lo que haga. O imagine que su cliente decide que una versión anterior del sitio estaba mejor. O que tiene que utilizar una versión anterior por motivos legales.

Imagine etre ~ase-en el que de& illteg,antes de 5\1 equipe de pregramaáén quieren trabajar con el mismo archivo. Puede que ambos hayan abierto y modificado el archi!O al mismo tiempo, sobrescribiendo los cambios. Puede que tengan una copia con la que trabajen localmente y la modifiquen de distinta forma. Si ha pensado en que todo esto puede suceder, puede que haya un programador sentado sin hacer nada mientras espera a que otro termine de modificar un archivo.

Todos estos problemas se pueden solucionar con un sistema de control de versiones. Estos sistemas pueden monitorizar los cambios de cada archivo de forma que no sólo podamos ver el estado actual de un archivo, sino también su aspecto en un momento concreto. Esta opción nos permite devolver un fragmento de código dividido a una versión que sabemos que funciona. Podemos adjuntar un conjunto de instancias de archivos concretos como versión de lanzamiento, lo que significa que podell\OS proseguir con el desarrollo del código pero con acceso a una copia de la versión de lanzamiento actual en cualquier momento.

Los sistemas de control de versiones también contribuyen a que varios programadores puedan trabajar conjuntamente en el código. Cada programador puede obtener una copia del código del archivo y, cuando realice algún cambio, se almacenan en el repositorio, o se confirman. De esta forma, los sistemas de control de versiones pueden saber quién ha realizado cada uno de los cambios en el sistema.

Estos sistemas suelen disponer de una función para administrar actualizaciones simultáneas, lo que significa quedos programadores pueden modificar el mismo archivo al mismo tiempo. Por ejemplo, imagine que John y Mary han sacado una

EL_HOLISTICO@HOTMAIL.COM - 2006

copia de la versión mú reciente del proyecto. John concluye sus cambios en un determinado archivo y los confirma. Mary tambifn realiza cambios en ese archivo e intenta confirmarlos. Si no han hecho los cambios en la misma parte del archivo, el sistema de control de versiones combinará las dos versiones del archivo. Si los cambios entran en conflicto, Mary recibirá una notificación y se le mostrarán las dos versiones. De esta ~ puede ajustar su versión del c6digo para evitar los conflictos.

El sistema de control de versiones utilizado por la mayoría de los programadores de Unix y/ o de código abierto es CVS (Sistema de versiones concurrentes). CVS es código abierto, viene incorporado prácticamente en todos los sistemas Unix y tambifn se puede ejecutar en DOS o Windows y Mac. Admite un modelo cliente-servidor para poder obtener o confirmar código desde cualquier equipo con una conexión a Internet, siempre que el servidor CVS se pueda ver en la red. Se utiliza para el desarrollo de PHP, Apache y Mozilla, "1tre otros proyectos principales, en parte al menos por esta razón.

Puede descargar CVS para su sistema desde la página de CVS en http://www. cv•home .org.

Aunque el sistema CVS base es una herramienta de linea de comandos, existen diversos complementos que le proporcionan una interfaz mú vistosa, incluyendo interfaces basadas en Java y Windows, a las que tambim se puede acceder desde la página de CVS.

Bitkeeper es otro producto de control de versiones, utilizado en importantes proyectos de código abierto que incluyen MySQL y Linux. Es gratuito para los proyectos de código abierto.

Hay alternativas comerciales a CVS. Una de ellas es perforce, que se ejecuta en la mayoria de las plataformas mú conocidas y es compatible con PHP. Aunque es comercial, se ofrecen licencias gratuitas para proyectos de código abierto en la direcdón http: / ／—. perforce. com.

Seleccionar un entorno de desarrollo

Al hablar del control de versiones hay que hacer referencia al tema de los entornos de desarrollo. Realmente basta con un editor de textos y un navegador para realizar las pruebas, pero los programadores suelen ser mú productivos en un entorno integrado o IDE.

Existen diversos proyectos gratuitos para generar un IDE PHP dedicado, incluyendo ICPHPDevelop, para el entorno de escritorio lCDE bajo Linux, que podrá encontrarenhttp: //kpbpdev.sourceforge.net.

Sin embargo, actualmente los mejores IDE PHP son todos comerciales. Zend Studio de zend. com, Komoclo deacti vea tate. com y PHPEd de nuspbere. com cuentan con IDE muy completos. Todo ellos disponen de una descarga de prueba pero es necesario adquirir el producto para continuar utilizándolo. Komoclo dispone de una licencia de uso no comercial muy económica.

Documentar nuestros proyectos

Podemos producir distintos tipos de documentación para nuestros proyectos de programación incluyendo, entre o~ los siguientes:

- Documentad6n cfte ctisel\o
- pocumentad6n ticnica/ gufa del programador
- Diccionario de datos (inclu~o documentación sobre clases)
- Gufa del usuario (aunque la mayoría de las aplicaciones Web deberían ser autoexplicativas)

Nuestro objetivo no es ensellarle a escribir documentación Ncnica sino sugerirle que lo haga todo mú Mal si automatiza parte del proceso.

En algunos lenguajes, existen diversas formas de generar automáticamente algunos de estos documentos, sobre la documentación h!cnica y los diccionarios de datos. Por ejemplo, javadoc genera un úbol de archivos HTML que contienen prototipos y una dacripción de los miembros de clases para programas Java.

Algunas utilidades de este tipo disponibles para PHP son las siguientes:

- phpdoc, disponible en http://www. phpdoc. de.

  Es un sistema utilizado por PEAR para documentar código. Debe saber que el lfflnino phpDoc se utiliza para describir varios proyectos de este tipo, de los cuales es uno.

- PHPDocuiaentor, disponible en bttp: / /pbpdocu. sourceforge. net.

  Genera un resultado muy similar al de j avadoc y parece que funciona correctamente. Tambim parece que cuenta con un equipo de programación mú activo que los otros dos.

- pbpautodoc, disponible en http://sourceforge.net/projecta/ pbpautdoc.

  Como en el caso anterior, produce un resultado similar al de j avadoc.

Si necesita mú aplicaciones de este tipo, (y componentes PHP en general), consulte Soun:eForge, en la dirección ht tp: / / sourceforge . net.

Lo utiliza búicamente la comunidad UNIX/Unux pero también hay numerosos proyectos para otras plataformas.

Prototipos

La creadón de prototipos es un delo vital de desarrollo utilizado habitualmente para diseñar aplicaciones Web. Un prototipo es una herramienta muy dtil para

EL_HOLISTICO@HOTMAIL.COM - 2006

satisfacer los requisitos del cliente. Normalmente se trata de una versión simplificada parcialmente funcional de una aplicación que se puede utilizar para comentar con el cliente y como base del sistema final. A menudo, las múltiples iteraciones de un prototipo generan la aplicación final. La ventaja de este enfoque es que nos permite trabajar junto al diente o el usuario final para producir un sistema con el que estén conformes y que, de algún modo, les pertenezca.

Para poder crear de forma conjunta un prototipo con rapidez, necesitará determinados conocimientos y herramientas. Y aquí es donde mejor funciona un enfoque basado en componentes. Si tiene acceso a un conjunto de componentes ya existentes, disponibles tanto de forma interna como públ~, podrá hacerlo con mayor rapidez. Otra herramienta de gran utilidad para el rápido desanollo de prototipos son las plantillas, que veremos en el siguiente apartado.

Existen dos problemas principales a la hora de utilizar el enfoque de prototipos. Tendd que conocerlos para poder evitarlos y asf utilizar este enfoque con todo su potencial.

El primer problema es que, por alguna razón, a los programadores les resulta diffcil desprenderse del código que han escrito. Los prototipos ie esaiben con rapidez y, con la ventaja del tiempo transcurrido, comprobará que no ha generado un piototipo de forma óptima. Las secciones extensas de código se pueden mejorar, pero 'si la estructura general no es correcta, entonces tiene un verdadero problema. El pro~lema es que normalmente las aplicaciones Web se generan bajo una gran pi:esión y en ocasiones no hay tiempo para corregirlas. De esta forma nos encontramds con un sistema pobremente diseñado que resulta difícil de mantener.

Tuda esto Se paede evitar con una derta phadfkact6n como MDIOs mencionado anteriormente:No obstante, recuerde que en ocasiones resulta más sencillo realizar urfboc:etoy empezar de nuevo que corregido. Aunque pueda parecerle quano tiene tiempo para ello, le ahorrad muchos sufrimientos posteriores.

El segundo problema de los prototipos es que un sistema se puede corwertir en un prototipo eterno. Cada vez que piensa que ha terminado su diente le sugiere nuevas mejoras o funcionalidades adicionales o actualizaciones del sitio. Todo esto puede provocar que no acabe el proyecto.

Para evitar este problema, disei\e un plan del proyecto con un número de i~ fijo asf como una fecha despuá de la que no se permita aftadir nueva funcionalidad sin planificar, programar y diseñar un presupuesto.

Separar lógica y contenido

Probablemente estf familiarizado con el uso de HTML para describir una estructura de documentos Web asf como de hojas de estilo en cascada (CSS) para describir su aRUiencia. Esta idea de separar la presentación del código se puede extender a la propamación. Por lo general, los sitios resultan más sencillos de utilizar y de mantener a largo plazo si separamos la lógica del contenido de la presenmc:ión, en definitiva, si separamos PHP y HTML

En proyectos sencillos con un número reducido de lineas de código, puede convertirse en un inconveniente más que en una ventaja. Cuando los proyectoecrecen, resulta fundamental encontrar la forma de separar lógica y contenido. Si no lo hace, su código será cada vez más diftcil de mantener. Si decide aplicar un nuevo diseAo a su sitio Web e incrusta gran cantidad de HTML en el código, el cambio del diseAo se convertirá en una pesadilla. Hay tres enfoques básicos a la hora de separar lógica y contenido, que describimos a continuación:

- Utilice archivos para almacenar distintas partes del contenido. Es un enfoque muy simple pero si el sitio es básicamente estático, funciona perfectamente. Este tipo de enfoque se explicó en un ejemplo de un capítulo anterior.
- Utilice un API de funciones o de clases con un conjunto de funciones miembro para conectar contenido dinálpico a plantillas de páginas estáticas. Tambiél describimos este enfoque en un capitulo anterior.
- Utilice un sistema de plantillas. Permite analizar plantillas estáticas y utilizar expresiones regulares para reemplazar etiquetas de marcadores de posición con datos dinámicos. La principal ventaja es que si otra persona diseña nuestras plantillas, como por ejemplo un diseñador gráfico, no es necesario que domine el código PHP. Podremos utilizar las plantillas proporcionadas con mínimas modificaciones.

Existen div4!rsos sistemas de plantillas disponibles. Probablemente el más conoddo sea Smar:ty, q,.e pHecle mconlAr en bttp I*I/WR>f,* pbp, -c.

Optimizar el código

Si provie~e de un entorno no relacionado con la programación Web, la optimización le resultará de gran importancia. Al utilizar PHP, la mayor parte del tiempo de espera de un usuario se debe a la conexión y descarga de una aplicación Web. La optimaciónde código apenas tiene efecto sobre estos tiempos de espera.

Utilizar una optimización sencilla

No obstante puede realizar algunas sendUas optimaciones que sí suponen una derta mejoria. Muchas están relacionadas con aplicaciones que integran una base de datos como MySQL en el código PHP, como indicamos a continuación:

- Reduzca las conexiones a bases de datos. La conexión a una base de datos suele ser la parte más lenta de cualquier secuencia de comandos, Jara evitarlo puede utilizar conexiones persistentes.
- Aumente la velocidad de la consultas de bases de datos. Reduzca el nwneio de consultas realizadas y asegúrese de optimizarlas. Con una consulta com-

EL_HOLISTICO@HOTMAIL.COM - 2006

pleja (y, por lo tanto, lenta), hay mú de una forma de hacer las cosas. Ejecute sus consultas desde la interfaz de lfnea de comandos de la base de datos y experimente con distintos enfoques para acelerar la operación. En MySQL, puede utilizar la instrucción BXPLAIR.para ver dónde puede estar el error de la consulta (en un-capitulo anterior se describió el uso de esta instrucción). Por I~ general, conviene minimizar las uniones y maximizar el uso de índices.

- Minimice la ¡eneraci6n **de contenido est6tico** desde PHP. Si **tocio el** HTML que genera proviene de echo o print (), tardar, mucho mú (es uno de los argumentos para cambiar hacia la separación de lógica y contenido que mencionamos antes). Tambifn se aplica a la generación dinúnica de botones de imagen; podemos utilizar PHP para generar los botones una vez y volver a utilizarlos cuando sea necesario. Si solamente genera p,ginas est4ticas a partir de funciones o plantillas cada vez que se carga una P'gina, puede pensar en ejecutar las funciones o en un utilizar las funciones una vez y en guardar el resultado.
- Utilice funciones de cadena en lugar de expresiones regulares siempre que sea posible. Son mú ,,pidas.
- Utilice cadenas de comillas simples en lugar de cadenas de comillas dobles siempre que sea posible. PHP evalúa las cadenas de comillas dobles y busca variables que pueda sustituir. Las cadenas de comillas simples no se evalúan. Por otra parte, si aparece entre comillas simples, seguramente se trate de contenido est,tico. Repase lo que estt! haciendo y compruebe si puede eliminar toda la cadena si la convierte a HTML esUtico.

Utilizar productos Zend

Zend Technologies es propietaria del motor de programación de secuencia de comandos de PHP (código abierto) que se ha utilizado desde PHP 4. Ad~s del motor búico, tambiffl puede descargar Zend Optimizer. Se trata de un optimizador multipasada que optimiza el código por nosotros y qlle puede a~tar la velocidad de ejecución de las secuencias de comandos desde un 40 a un 100por den. Para ejecutar el optimizador necesitar, PHP 4.0.2 o una versión superior. Aunque se trata de código cerrado, se puede descargar gratuitamente desde el sitio de 2.end, http://www.zend.com. El funcionamiento de este complemento consiste en optimizar el código generado por la compilación en tiempo de ejecución de nuestra secuencia de comandos. Entre otro& productos de Zend podemos destacar Zend Studio, Zend Accelerator, Zend Encoder así como acuerdos de compatibilidad comercial.

Pruebas

La revisión y la comprobación del código es otro aspecto Wsico de la ingeniería de software que se suele p~ por alto en programación Web. Es muy habitual

tratar de ejecutar el sistema con dos o tres pruebas, y decir que funciona conectamente, un error muy común. Compruebe que ha revisado y probado exhaustivamente distintos casos antes de preparar la producción del proyecto.

IA! sugerimos dos enfoques que puede utilizar para reducir los errores de ~ código (nunca se pueden eliminar de forma conjunta, pero si es posible eliminar o minimizar la mayoría de ellos).

En primer lugar, adopte una pr'ctica de rwisión de código. Es el proceso mediante el que otro programador o equipo de programadores revisan el código y sugieren mejoras. Mediante este tipo de aMlisis se sugiere lo siguiente:

- Errores que ha pasado por alto
- Comprobaciones sobre las que no ha pensado
- Optimización
- Mejoras de la seguridad
- Componentes existentes que puede utilizar para mejorar un &agmento de código
- Funcionalidad adicional

Incluso si trabaja por su cuenta, es aconsejable encontrar un •cotega program~-dor" que estt! en la misma situación y revisar el código entre ambos.

La segunda sugerencia consiste en localizar personas que prueben nuestras aplicaciones Web y que representen a los usuarios finales del producto. La principal diferencia entre aplicaciones Web y aplicaciones de escritorio es $lue cualquiera puede utilizar una aplicación Web.

No debe asumir que los usuarios estarm familiarizados con la info~tica. No es aconsejable proporcionales un extenso manual o una mínima tarjeta de referencia. Por el contrario, las aplicaciones Web se deben documentar y explicar por si m——. Debe pensar en cómo utWzarm los usuarios la aplicación. La capacidad de uso tiene una importancia fundamental.

Resulta muy complicado comprender los problemas a los que se enfrentan los usuarios noveles si somos experimentados programadores o usuarios de la Web. Una forma de solucionar este aspecto consiste en recurrir a gente que represente a usuarios convencionales.

En el pasado, sólo se publicaban aplicaciones Web en versiones beta para solucionar este problema. Cuando crea que ha corregido la mayoría de los errores, ofrezca la aplicación a un reducido grupo de usuarios de prueba y acepte un tr6fico reducido en el sitio. Puede o&ecer servicios gratuitos a los 100primeros usuarios a cambio de información sobre el sitio. Le garantizamos que obtendn alguna combinación de datos o de usos sobre la que no había pensado. Si tiene que genenr un sitio Web para una empresa cliente, &ta puede proporcionarle un grupo de usuarios noveles y hacer que sus empleados trabajen en el sitio (es una de las ventajas implícitas de fomentar la sensación de propiedad del producto por parte del cliente).

EL_HOLISTICO@HOTMAIL.COM - 2006

Lecturas adicionales

Exiallegrancmtidad ele ma•rial sola este útaque podñmnoaclelCribir: bemol lwcho **mcla a lacimda de la inpnilda desoftwme.solne loque 1emnwrito ffllllll"OW .....

Una obra que explica la dicoloaúa entre sit.Web-c:omo-docwnmy li~ Web-como-apticacidn es *Ww* Site *En,-riaf:* ae,-, *W,1,* Pllp Dcsip de Thoam A. Powell. Cualquier libio Nbft ingpierfa ele IOftwue le Hffid comoremenda.

Si MCllita mú inf'onnaci6n sobn el c:cnbol de fflSi6n puede visitarel sitio Web de CVS, bttp: //www.cvsboe.org.

No existen muchaa publicaciones sola c:cntrol ele versidn (lo que resalta 11>r-prmdmlle debido a su importmcia) peio puede praNrcono,,,. *Soura* Dnelop,rrfflt wlt *CVS* de lCad Pranz Pogel o *CVS Podrd* Rlfomra de Gregw N. Purdy.

En SomcePorge encontrad componentes PHP, IDE o sistellla5 ele docmnenta-ci6n, http: //•ourceforge. net.

M~ de loe teallIS analizadoe en este capftalo • deKriben en artkuloa del sitio ele 7.end, dande MgIll'llmente encantrañ ms inbmaci6n al 1espec10. Tam-.,. ..... el opliurizadui1delsitio;WW',nnd~c:aa

~ .- cipftalo le ha parecido intensante, no duele en visitar Extreme PwP.~w:a IMIDdode-desarrollocle software dirigido a dominma loa que IQJ~con ~ como es el calO deldesarmUo Web. Ehitio

~W,-.l ;;_de~JlatNiit~~-~~~P~z-np~·~mt-ig~--~.bt~tp~~l_l_..,_:_·_eixt~z~-'~'-'•_'_--~t-ll!J~~ª-,:g:_._~~~

En este capítulo analizaremos la depuración de secuendas de comandos de PHP. Si ha realizado los ejemplos del libro o ha utilizado PHP antes, seguramente habrá desarrollado sus propias tfcnicas y conocimientos de depuración. Al aumentar la complejidad de nuestros proyectos, la depuración tambiét se complica. Aunque mejoremos nuestros conocimientos, es muy probable que los errores impliquen un mayor número de archivos o de interacciones entre el código de varias personas.

En este capítulo abordaremos los siguientes temas:

- Tipos de errores de programación
  - Errores sintácticos
  - Errores de ejecución
  - Errores lógicos
- Mensajes de error
- Niveles de error
- Desencadenar errores propios
- Solucionar errores con elegancia

Errores de programación

Independientemente del lenguaje que utilice, existen tres tipos generales de errores de programación:

EL_HOLISTICO@HOTMAIL.COM - 2006

- Errores sintácticos
- Errores de ejecución
- Errores lógicos

Veremos cada uno de estos tipos antes de analizar las técnicas para detectar, procesar, evitar y solucionar errores.

Errores sintácticos

Los lenguajes disponen de una serie de normas denominada sintaxis que las instrucciones deben cumplir para que sean válidas. Esto se aplica tanto a los lenguajes naturales, como el espai\ol, como a los lenguajes de programación, como PHP. Si una instrucción no cumple las normas de un lenguaje, se dice que se ha producido un error sintáctico. También se suelen denominar errores de analizador cuando se habla de lenguajes interpretados como PHP o errores de compilador, cuando se trata de lenguajes compilados como C o Java.

Si no cumplimos las reglas sintácticas de un idioma, por ejemplo del español, es probable que la gente entienda lo que queremos decir, algo que no suele pasar con los lenguajes de programación. Si una secuencia de comandos no cumple las reglas sintácticas de PHP, es decir, si contiene errores sintácticos, el analizador de PHP no podrá procesarla. Los humanos podemos obtener información de datos parciales o en conflicto. Los ordenadores no.

Entre otras muchas reglas,. la sintaxis de PHP requiere que las instrucciones tenniriin en punto y coma, que las cadenas se incluyan entre comillas y que los parámetros pasados a funciones se separen con comas y se encierren entre paréntesis. Si incumplimos estas normas, es improbable que nuestra secuenci~ de comandos de PHP funcione y es muy probable que genere un mensaje de error la primera vez que intentemos ejecutarla.

Una de las principales ventajas de PHP son los útiles mensajes de error que indican que algo no funciona correctamente. Un mensaje de error de PHP nos indica qué ha salido mal, en qué archivo se ha producido el error y en qué línea se encuentra.

Un mensaje de error se parece a lo siguiente:

```
Par•• error: par•• error in
/holle/baok/public_html/chapter25/error.phpon line 2
```

Este error lo produjo esta secuencia de comandos:

```
c?php
   $date•  date(a.~.y');
?>
```

Apreciará que hemos intentado pasar una cadena a la función date (l pero que accidentalmente hemos olvidado la comilla de apertura que indica el inicio de la

cadena. Los errores sintácticos sencillos como éste son los más fáciles de localizar. Podemos cometer un error similar, pero más difícil de localizar, si olvidamos terminar la cadena, como se muestra en este ejemplo:

```
c?php
   $date• date(•m.d.y);
```

Esta secuencia de comandos generará el siguiente mensaje de error:

```
Par•• error: per•e error in
/lu>me/book/public_htal/cllapter25/error.pJapn line 4
```

Evidentemente, como nuestra secuencia de comandos sólo tiene tres líneu, nuestro error no se encuentra realmente en la cuarta•

Los errores en los que abrimos algo pero olvidamos cerrarlo se muestr~n de esta forma. Este tipo de problemas aparecen al utilizar comillu simples y dobles, y también con las distintas formas de paréntesis.

La siguiente secuencia de comandos generará un error sintáctico similar:

```
c?php
   if  (true)
   {
      echo •aqui va el error';
?>
```

Resulta complicado localizar estos errores si se generen a partir de una combinación de varios archivos o si se producen en un archivo de gran tamaño. La aparición de .~rse error. on line 1001• en un archivo de 1.000_líneas puede ser sufiaente para arrumamos el día y nos indica que deberiamos intentar escritiir código más modular. Por lo general, los errores sintácticos son los más sencillos de localizar. Si comete un error sintáctico, PHP le mostrará un mensaje de error en el que indica dónde lo puede encontrar.

Errores de ejecución

Los errores de ejecución pueden resultar difíciles de detectar y corregir. Una secuencia de comandos contiene un error sintáctico o no lo contiene. Si lo tiene, el analizador lo detecta. Los errores de ejecución no sólo se producen a causa de los contenidos de la secuencia de comandos. Pueden deberse a interacciones entre las secuencias de comandos y otros,eventos o condiciones.

La siguiente instrucción:

```
require  ('~redearchivo.php•lu
```

es un argumento de PHP perfectamente válido. No contiene errores sintácticos.

Si~ embargo, puede g~erar un error de e;ecución. Si ejecuta esta instrucción y no exJSt~nombredearch1 vo. php o el usuano con el que se ejecuta la secuencia de comandos no tiene permiso de lectura, obtendremos un error como éste:

EL_HOLISTICO@HOTMAIL.COM - 2006

```
l'aa1 __ ɪ, —in()  (function.requlra),  l'ailed openiDf raqulred
'no.ibredearclaivo.pbp'
(lnclude_path•'.:/uar/local/llb/pbp'I   in
/bama/booJt/pub1ia_hta1/a~erll/arror.php on line 1
```

Aunque no hay nada erróneo en nuestro código, como depende de un archivo que puede o no existir en distintos mom~tos cuando se ejecute el código, puede generar un error de ejecución. Las tres siguientes instrucciones de PHP son v'1idas. Desafortunadamente, combinadas!intentan realizar algo imposible: dividir por cero.

```
U•   10,
$j  •  0;
$k  •  U/$k1
```

Este fragmento de código genera la siguiente advertencia:

```
W.rai119: Dlvialon by H:ro in
/IMl9a/lMtok/pub1ia_bta1/ahaptarll/div0.php on lin•  3
```

Resulta muy sencillo de corregir. Nadie escribiría a propósito un código que intentara dividir por c:ero, pero si no se comprueban las entradas de usuario se puede producir este tipo de error.

El siguiente código genera en ocasiones el mismo error pero puede resultar mú difícil de aislar y corregir ya que sólo se produce en ocasiones:

```
U•   10;
tk  •  $l/$_RBQUSST(•input•)1
```

Se trata de uno de los muchos errores de ejecución que podemos ver al probar nuestro código. Entre las principales causas de los errores de ejecución destacamos las siguientes:

- Llamadas a funciones que no existen.
- Lectura o escritura de archivos.
- Interacción con MySQL u otras bases de datos.....
- Conexiones a servicios de red.
- Comprobación de entrada de datos incorrecta.

Veremos cada una de estas causas individualmente.

Llamadas a funciones que no existen

Es muy habitual invocar funciones que no existen. Las funciones incorporadas suelen utilizar nombres sin excesiva coherencia. Por ejemplo, strip_tags (I tiene un guión bajo mientras que stripslaahes () no lo tiene.

Tambü!n es muy comlin invocar una función que no existe en la secuencia de comandos actual, pero sf en alguna otra parte. Si su código contiene una llamada a una función que no existe, como por ejemplo

```
nonaxiatant_function(I;
```


o

```
aiapelad_function(ll
```

obtendnl un mensaje de error similar al siguiente:

```
l'atal. error: Call to undefinad f,anction,  nonexiatant fUDCtion(I
in /~/IIOOk/pubUa_hta1/allapurH/error.pl,poa Une  1
```

Del mismo modo, si invoca una función que existe pero lo hace con un ndmero incorrecto de pañmetros, recibinl una advertencia.

La función stratr (I requiere dos cadenas: un pajar en el que buscar y una aguja que encontrar. Si por el contrario la invocamos de este modo:

```
atratrll,
```

Obtendremos la siguiente advertencia:

```
wanaiDfr•  Wrong par—tar count for atratrll in
/IIGme/llOOk/pubUa_hta1/ahapterH/error.pbpon Une  1
```

Como PHP no permite la sobrecarga de funciones, esta lfnea p pre ser6 incorrecta aunque no siempre veremos esta advertencia. La misma igfrucción dentro de la siguiente lfnea de comandos tambWn es incorrecta:

```
c?pbp
   if($var  •  •  41
   {
      at:r;atr(I,
   }
?>
```

pero a excepción del raro caso en que la variable $var tenga el valor 4, la invocación de stratr (1 no se produdnl, por lo que no se mostrar6 la advertencia. El intfrprete de PHP no se molesta en analizar secciones del c6digo que no son necesarias para la ejecución actual de la secuencia de comandos. Tendnl que comprobarlo minuciosamente.

La invocación incorrecta de funciones es algo muy habitual pero como los mensajes de error resultantes identifican la lfnea exacta y la llamada que origina el problema, se pueden corregir fácilmente. Solamente resultan diffáles de encontrar si el proceso de prueba es pobre y no comprueba todo el c6digo ejecutado condicionalmente. Al realizar pruebas, uno de los objetivos consiste en ejecutar todas las lineas de código una vez. Otro es probar todas las condiciones y clases de entrada.

Lectura o escritura de archivos

Aunque cualquier cosa puede salir mal en alglin momento de la vida ótil de un programa, algunas son m6s probables que otras. Los errores de acceso a archivos se suelen producir tan a menudo que es necesario procesarlos correctamente. Los discos duros fallan o se llenan, y los errores humanos hacen que cambien los directorios de permisos.

EL_HOLISTICO@HOTMAIL.COM - 2006

Funciones como fopen (l, que suelen fallar a menudo, normalmente devuelven un valor para indicar que se ha producido un error. En el cuo de fopen {l, el valor false indica la presencia de algún fallo.

En funciones que indican la presencia de un error, es necesario revisar con atención el valor devuelto por cada llamada.

Interacción con MySQL u otras bases de dalos

La conexión a MySQL y el uso del DIÍSDIO puede generar muchos errores. Por sí misma,la .funciónmysqli_connect (l puede generar al menos los siguientes errores:

- Warning: mysqli_connect (l [function.aysqli-ªconnect] : Can' t connect to MySQL server on 'localaost• (10061)
- Waming: mysqli_connect () [function.mysqli-connect) : Onknown MySQL Server Hoat 'bostname• (11001)
- Waming: mysqli_connect (l [function.myaqli-connect): Access denied for user: •username•e•localhoat• (Osing password: ¡YBS)

Como habr6 imaginado, mysqli_connect (l devuelve el valor falH cuando se pr&luce un error. Esto significa que este tipo de errores se puede detectar y corregir f,cilmente. Si no detiene la ejecución regular de su secuencia de comandos y procesa estos errores, la secuencia de comandos continuar, en su intento de eeae,8'n a la IJue de 4atoe Si iRleAta ejecutar c:oaaultu y obtmer raultados sin. una QJaexión MySQL v'1ida, los visitantes verm una pantalla repleta de mensajes de ern>r.. algo poco profesional.

Otras muchas funciones PHP relacionadas con MySQL como mysqli_query (l también devuelven false para indicar que se ha producido un error.

Si se produce un error, puede acceder al texto del mensaje de error por medio de la función .. yqli_error () oa un código de error por medio de—yqli_errno O. Si la última función MySQL no ha generado un error, myaqli_error (l devuelve una cadena vada y msqyli_errno (l devuelve o.

Por ejemplo, imagine que se ha conectado al servidor y ha seleccionado una base de datos. El siguiente &agmento de código:

```
$r••ult • ay9qli_qu•ryl$db, '••l•ct • froia dae•_not_•xi•t' ),
mc:bo ay9111i_•rrnol$db);
ec:llo •cbr j,.,
mcbo IIYIIC11i_emrl$db);
```

puede dar como resultado:

```
114'
Tallle 'dbnaM.doe• _not_mxi•t• doe•n't exi•t
```

Ffjese en que el resultado de estas funciones hace referencia a la última función MySQL ejecutada (que no sea mysqli_error () o mysqli_errno O). Si quiere

conocer el resultado de un comando, compru~lo antes de ejecutar otras. Al igual que los errores de interacción de archivos, también se producen errores de interacción de bases de datós. Incluso después de completar el desarrollo y la prueba de un servicio, deecubriñ que el demonio MySQL (mysqld) se ha colapsado o se ha quedado sin conexiones disponibles. Si su base de datos se ejecuta en otro equipo físico, depender6 de otro conjunto de componentes de hardware y de software que pueden fallar, es decir, otra red, otra conexión, otra tnjeta de red, otros enrutadores, etc., entre el servidor Web y el equipo de base de datos.

Tendr6que comprobar si las solicitudes de su base de datos se cumplen antes de intentar utilizar el resultado. No tiene sentido intentar ejecutar una consulta si no se puede establecer una conexión a la base de datos, ni tiene sentido extraer y procesar los resultados tras ejecutar una consulta fallidL

Debe saber que existe una diferencia entre una consulta fallida y una consulta que simplemente no puede devolver datos o afectar a ninguna fila.

Una consulta SQL que contenga errores sint6cticoeSQL o que haga referencia a bases de datos, tablas o columnas que s(existen puede fallar, por ejemplo la siguiente consulta:

```
••l•ct • frcl11 doe•_not_mxi•tl
```

falla porque el nombre de la tabla no existe y genera un número y un mensaje de error que se pueden recuperar con mysqli _errno (l y mysqli_error ()•

Una consulta SQL sintllcticamente v'1ida y que haga referencia solamente a bases de datos, tablas .y columnas que existan no fallará. Sin embargo, puede que no de''ieha resultados si se lealiz-a en mia tabla ,ada o busca dato.. que 110 existe.,, Imagine que se ha conectado correctamente a una base d~ datos y que tiene una tabla con el nombre tl y una col~ ce>n el no~ ~l.~ si~en~ COl18ult~:

```
••lmc:t • fram tl wun el • •not in d•tmti.••' ,
```

será satisfactoria pero no devolver, ningún resultado. Antes de utilizar el resultado de la consulta tendrA que comprobar si produce algún error *y* si devuelve resultados.

Conexiones a redes de servicios

Aunque los dispositivos y otros programas de su sistema pueden fallar ocasionalmente, apenas lo har'8 a menos que la calidad no sea la c:orrecta. Al utilizar una red para conec:tarse a otros equipos y al software de dichos equipos, tendrA que aceptar que parte del sistema fallará. Al conectarse de un equipo a otro, depende de numerosos dispositivos *y* servidos que no est6n bajo su control. Aunque seamos repetitivos, debe comprobar el valor devuelto por lu funciones que intenten actuar con un servicio de red. Una invocación de función como la siguiente

```
S•p • f•ockopen l 'loc•lllo•t', sooo ll
```

generará una advertencia si no puede conectarse al puerto 5000 del equipó localhost, pero la mostrará en el formato predeterminado y no permitirA a la

EL_HOLISTICO@HOTMAIL.COM - 2006

secuencia de comandos solucionarlo de forma elegante. Si volvemos a esctibir la invocación como

```
S•p • •f•ockopen ( 'localbo•t', 5000, aserrorno, aSerror•tr );
if (1$ap)
   ecbo "Bllll0R: Serrorno, Serroratr•;
```

se suprime el mensaje de error incorporado, se comprueba el valor devuelto para ver si se haproducido un error y se utiliza el propio código para solucionar el mensaje de error. Al escribir el código, mostrará un mensaje de error que puede ayudarle a solucionar el problema. En este caso, se generarla el siguiente resultado:

```
BRROR, 10035: A non·blocking aocket operation could not be C0llpleted
i-diately.
```

Los errores de ejecución son más diffciles de eliminar que los errores sint6cticos Yl! que el analizador no puede detectar el error la primera vez que se ejecuta el código. Como los errores de ejecución se producen como respuesta a una combinación de eventos, su detección y resolución puede resultar complicada. El analizador no puede indicar automáticamente que una linea determinada generaral un error. Las pruebas que realice deben tener en cuenta una de las situaciones que generen el error. El procesamiento de errores de ejecución requiere una cierta planificación: comprobar los diferentes tipos de errores que pueden producirse y, tru ello, emprender la acción adecuada. También es necesario probar todas lu clues de errores de ejecución que puedan producirse.

Esto no significa que tenga que simular todos y cada uno de los distintos errores que puedan producirse. Por ejemplo, MySQL puede proporcionar uno de aproximadamente 200 números y mensajes de error diferentes. Tendral que simular un error en todas lu invocaciones de función que puedan generar errores u(como un error para cada tipo que se procese por un bloque de código düerente.

Comprobar incorrectamente los datos de entrada

A menudo uumimos la validez de los datos quejntroducen los usuarios. Si estos datos no coinciden con nuestras expectativas, pueden generar un error, bien de ejecución o bien lógico (que analizaremos en el siguiente apartado).

Un ejemplo clalsicode error de ejecución se produce cuando se procesan datos introducidos por los usuarios y olvidamos aplicar addslashea () a los mismos. Esto significa que si tenemos un usuario con el nombre O'Grady que contiene un apostrofe, obtendremos un error de la función de base de datos.

En el siguiente apartado, describiremos con más detalle estos errores provocados a causa de suposiciones sobre datos introducidos.

Errores lógicos

Los errores lógicos son el tipo de error mú dificil de localizar y eliminar. Este tipo de error se produce cuando un código perfectamente vallido hace lo que se le

dice, pero no lo que el programador esperaba. Los errores 16gicosse pueden producir a causa de un simple error órtográfico, como por ejemplo:

```
for <Si• 0, Si e 10, Si++>,
(
   acbo 'doing ao.etbingcbr t~•;
```

Este fragmento de código es perfectamente vallido. Cumple correctamente la sintaxis de PHP. No depende de servicios externos, por lo que es poco probable que falle durante el tiempo de ejecución. A menos que lo haya revisado minuciosamente, es probable que no haga lo que pensamos o lo que el programador espera.

A simple vista, parece que iterará diez veces el bucle for, repitiendo •doing something • cada vez. La inclusión de un punto y coma al final de la primera linea significa que el bucle no afecta a las lineas siguientes. El bucle for iterará diez veces sin resultados y, tras ello, la instrucción echo se ejecutaraluna vez.

Como este código es una forma perfectamente válida pero ineficaz de escribir código para obtener este resultado, el analizador no se queja. Los ordenadores son muy inteligentes para algunas cosas pero carecen de sentido comdn o de inteligencia. Un ordenador hace exactamente lo que se le dice. Tendralque asegurarse de que lo que le dice es exactamente lo que necesita.

Los errores lógicos no se deben a ningún fallo del código, sino a un fallo del programador que ha escrito el código que indica al ordenador lo que debe hacer. Como resultado, los errores no se pueden detectar automalticamente. No sabremos que se ha producido un error y no tendremos un número de linea en el que buscar el problema. Los errores lógicos sólo se detectan por medio de las pruebas adecuadas.

Un error lógico como el del ejemplo anterior se puede cometer con facilidad, pero también resulta sencillo de corregir ya que la primera vez que se ejecute el código obtendremos un resultado distinto al esperado. La mayorfa de los errores 16gicosson ligeramente mú nocivos.

Los errores lógicos problemáticos suelen deberse a suposiciones erróneas de los programadores. En el capftulo anterior recomendamos la colaboradón de otros programadores para revisar nuestro código y sugerir casos de prueba adicionales asf como acudir a nuestro público de destino en lugar de a programadores para realizar las pruebas. La suposición de que los usuarios sólo introducen determinados tipos de datos es muy habitual y también es muy posible pasarlo por alto si realizamos nuestras propias comprobaciones.

Imagine que disponemos de un cuadro de texto para pedidos en un sitio de comercio. ¿Ha supuesto que todos los usuarios han introducido únicamente valores positivos? Si un usuario introduce -10, ¿el software cargará en la tarjeta de mdito del usuario diez veces el precio del artículo?

Imagine que tiene un cuadro para introducir una cantidad en euros. ¿Los usuarios pueden introducir la cantidad con y sin el símbolo del euro? ¿Pueden introducir cifras con millares separadas por puntos? Algunos de estos upectos se pueden comprobar en el lado del cliente (por ejemplo por medio de JavaScript) para aliviar la carga de nuestro servidor.

EL_HOLISTICO@HOTMAIL.COM - 2006

Si puamos información a otra página, puede que haya pensado que en la cadena que estamos pasando existen caracteres con un significado especial en un URL como los espacios.

El número posible de errores lógicos es infinito. No hay una forma automática de comprobar todos ellos. La única solución consiste, en primer lugar, en intentar eliminar suposiciones que implícitamente hayamos codificado en la secuencia de comandos y, tras ello, probar con todos los tipos de entradas válidas y no válidas posibles, asegurándonos de que obtenemos el resultado anticipado de todos ellos.

Ayuda de depuración de variables

Al aumentar la complejidad de los proyectos, puede resultar muy útil disponer de algún código de ayuda para identificar el origen de los errores. En el listado 25.1 incluimos un fragmento de código que puede encontrar de utilidad. Este código duplica los contenidos de las variables pasadas a nuestra página.

Uatado 2s.1; dump:_var1ables.php. Este código se puede Incluir en ~lnas p.a volcar los contenidos de las varlables para la depuración.

```
e?pbP
 :[/  eecaa linea• aplican el foraaco de comencarioa HTML al reaulcado
  //  e in~an dump_array repecida..nce

 ~--b0~+-l•\"n_        IIIJ;ftM-1  '•'AltlldltdtdUNP···~
  '''
  fllho 'el--  BBGIR GBT VARS -->\D'1
  echo 'e !--  , .dump_array($_GBTI. ,  ~-:i.\n•  ī

  ecbo 'e!--  BBGIR POST VARS -->\n';
  echo :el--  '.dump_array($_POSTI.'   -->\D';

  echo 'e!--  BBGIR SBSSION VARS -->\D';
  echo 'el--  '.dump_array|$_SBSSION).'   -->\n';

  echo 'el--  BBGIR COOKIB VARS -->\D';
  echo •et--  '.dump_array($_COOKIB|.'   -->\n';

  echo '\ne !-- BIID VARIABLB DUMP -->\n';

// dump:_array() uciliza la funci6n incorporada princ_r
// y eacapa codo• loa comencarioa RTML final••

funccion dump_array($array1
(
   $oucpac • prin~_r($array, cruel;
   soucpac • acr_replace('-->',   •--  >',  $oucpucl,
   recuna' Soucpuc,


?>
```


Este código representa cuatro matrices de variables que recibe la página. Si la página se ha invocado con variables GET, variables POST,cookies o tiene variables de sesión, dará como resultado todos estos elementos.

Tendnmos que incluir el resultado dentro de un comentario HrML para que se pueda ver, pero que no interfiera con la forma en la que el navegador representa los elementos de página visibles. Se trata de una buena IKnica para generar información de depuración. Al ocultar la información de depuración en comentarios, como hemos hecho en este código, podemos dejar el código de depuración hasta el último momento. Hemos utilizado la función dump_array() como contenedor para print _r ( ) . La función dump_array ( ) simplemente escapa todos los caracteres de comentarios HTML finales.

El resultado exacto dependerá de las variables pasadas a la página, pero si lo ailadimos a uno de los ejemplos de autenticación de un capitulo anterior, incl_uye las siguientes lineas al HTML generado por la secuencia de comandos:

```
e1-- 8BGI1' VARIABLE DUMP -->

e!--  BBGIR GBT VARS -->
e1--  Array
(
1


el--  8BGI1' POST VA11S -->
el-   Array
(
   (uaerid]  •> ceacuaer
   (paaawordl  •> paaaword

-->
el--  BIGIR SBSSI0Jr VARS -->
el--  Array
(
1
-->
el--  BBGIR COOKIB VJU1S -->
e!--  Array
(
   [PHPSBSSIDJ  •> ~2b5f55fad985dd73af33f470f3C1855

·->

e1--  1111> VARIABLB .D0NP - ->
```

Apreciará que muestra las variables POSTenviadas desde eHormulario de inicio de sesión de la página anterior: user id y password. También muestra la variable de sesi6n que utilizamos para almacenar el nombre del usuario: valid_user. Como vimos en up capítulo anterior, PHP utiliza una cookie para vincular variables de sesión a determinados usuarios. Nuestra secuencia de comandos duplica el número pseudo aleatorio, PBPSESSID, que se almac:ena en dicha cookie para identificar a 1m determinado usuario.

EL_HOLISTICO@HOTMAIL.COM - 2006

Niveles de informes de errores

PHP nos permite especificar la claridad de los errores. Podemos modificar los tipos de eventos que generar6n lI\énlajes. I5é fonna predeterminada, PHP infoTma de todos los errores que no sean avisos.

El nivel de informes de error se establece por medio de una serie de constantes predefinidas, como se indica en la tabla 25.1.

T.._ 21.1. CoNIIlntea de Informas de emns.

Valor de operación	Significado	
1	B_BRROR	Informa de errores graves del tiempo de ejecu-cl6n
2	B_lfARNING	Informe de errores no graves del tiempo de ejecucl6n
4	B_PARSB	Informa de errores de an6ffsls
8	B_NOTICB	Informe de avisos, notiftceclones de que algo que hemos hecho es un error
18	B_CORB_BRROR	Informa de fallos al InlclarM el motor de PHP
32	B_CORB_lfARNING	Informe de fallos no graves durante e!1ñ1c10 del motor de PHP
84	B_COMPILB_BRROR	Informa de errores de cornpilacl6n
128	B_COMPILB_lfARNING	Informa de errores de compllacl6n no graves
258	B_OSBR_BRROR	Informa de errores desencadenados por el uaua-rlo
512	B_OSBR_lfARNING	Informa de adY9r'1!nclas d... ncadenadas por el usuario
1024	B_OSBR_NOTICB	Informa de avilos desencadenadoe por el usua-rlo
2047	B_ALL	Informa de todos los errores y advertencias
2048	B_STRICT	Informa del uso de un comportamiento obsoleto y no rec:omendado; no se Incluye en B_ALL pero resulta muy IItll para modlflcar el Tac:tor del c6dlgo

Cada constante representa un tipo de error que se puede mencionar o ignorar. Por ejemplo, si especificamos el nivel de error B_BRIIOR, solamente se informará de errores graves. Estas constantes se pueden combinar por medio de aritmética binaria

para generar distintos niveles de error. El nivel de error pri:deteamlnado, el informe de todos los errores que no selin avisos, se especifica como

B_ALL • -s_mnc:s

Esta expresi6n esti fonnada por dos de las constantes predefinidas combinadu con operadores arillMticos en orden de bils. El símbolo *t.* es el operador ANO en orden de bils y el sfmbolo - el operador NOT en orden de bits. Esta expresión se puede leer como B_ALL AIID NOT B_NOTICB,

B_ALL es, por sf mismo, una combinación de todos los tipos de error restantes, menos B_STRICT. Se podría sustituir por los otros niveles combinados por medio del operador OR en orden de bils (|).

B_BUOR | 1_... 1m | I_PUSB | I_IIOTICB | B_COU_IIUIOJI | 1_cou_wu111m | I_CONPILI_IRIOR 11_CICINPILB_IIABDO 11_usBR_BIIIIOII | I_UIBR_alIlrDG | B_usn_IIOTICI

Del mismo modo, el nivel de informe de error predeterminado se podría especificar por todos los errores a excepción de avisos combinados con OR.

B_IUOR | B_wunm | I_PARSB | I_CORB_IRROR | a_coRB_IIAIIIIIm | B_CONPILI_BRROR | B_CONPILB_11A11111m | B_USBR_SRROR | B_usaa_WAIIJI?m | a_USBR_IIOTICS

Modificar los parámetros de informes de error

Podemos configurar los par6metros de informes de error globalmente en el ar·chivo php. ini o por secuencia de comandos.

Para modificar el informe de errores en todu las secuencias de comando, puede cambiar estas cuatro líneas en el archivo php. ini predeterminado:

```
error_reporting •      B_ALL • -B_IIOTICB
diqilay_arrorm         on
log_arrors             ou
track_arror•           Off
```

Los parámetros globales predeterminados son para

- Informar de todos los errores menos de·avisos
- Mostrar mensajes de error como HTML en resultado estindar
- No registrar mensajes de error en disco
- No realizar el seguimiento de errores y almacenar el error en la variable

El cambio que seguramente realice es convertir el nivel de informe de errorés hasta B_ALL | B_STRICT. Esto provocar, que se informe de numerosos avisos, de incidentes que pueden indicar un error o pueden deberse a que el programador se aproveche de la d.a,il naturaleza de PHP y del hecho de que aotom6ticamente inicializa las variables a o.

EL_HOLISTICO@HOTMAIL.COM - 2006

Durante la depuración, puede resultarle útil establecer un nivel error_ reportingsuperior. En código de producción,si proporciona mensajes de errores propios, resulta más profesional desactivar di•play_error• y activar log_errors, a la vez que se mantiene el nivel error_reporting con un valor elevado. De esta forma podrá hacer referencia a errores detallados en los registros si aparece algún problema.

Al activar track_error• podemos controlar errores en nuestro propio código, en lugar de dejar que PHP nos ofrezca su funcionalidad predeterminada.Aunque PHP cuenta con mensajes de error muy útiles, su comportamiento predeterminado no es agradable cuando algo sale mal.

De forma predeterminada, cuando se produce un error grave, PHP ofrece el siguiente resultado:

```
cbr>
cb>2'l.PO  d•  errorc/b>:   —a•aje  de  error   ea  cb>rul:a/arc:!l.ivo.pbpc/b>
eD  llaea   cb>ad.  roL!aeac/b>cbr>
```

y detiene la ejecución de la secuencia de comandos. Si se trata de errores no graves, se muestra el mismo texto, pero se permite que prosiga la ejecución.

liste resultado HTML resalta el error, pero su aspecto es pobre. Es poco probable que el estilo del mensaje de error encaje con el del sitio. Puede que los usuarios de Netscape no vean ningún resultado si el contenido de la página se muestra dentro dt!" una tabla. Esto se debe a que el HTML que abre pero no cierra elementos de tabla, como por ejemplo

```
ctúle>
ctr>ctd>
--cbr>-
cb>2'l.PO  d•  errorc/b>:   -aje      de  error   ea  cb>rul:a/arc:!11vo.pllpc/b>
ea  llaea   cb>lld.  roLlaeac/b>cbr>
```

se representa en forma de pantalla en blanco en alguru,e navepdores.

No es necesario conservar el comportamiento de procesamiento de errores predeterminado de PHP ni siquiera utilizar los mismos parámetros en todos los archivos. Para cambiar el nivel de informe de errores en la secuencia de comandos actual, podemos invocar la función error_reporting ().

Al pasar una constante de informe de error o una combinación de constantes, se establece el nivel de la misma forma en la que lo hace la directiva similar en php. ini. La función devuelve el nivel de informe de error anterior. Una forma habitual de utilizar esta función es la sigui~nte:

```
// de.active  el  iafor11ede  error••
Sold_l.,,.l    • error_reporti119COI;
//  aa.da  equi  el  código  qua  g—ra  adYertaaciu
il  ,,..l•a  a  activar   al  iDfor11eda  errorea
e:rror_l:911ortingCSold_level);
```

Este fragmento de código desactiva los informes de error, lo que nos permite ejecutar código que pueda generar advertencias que no queremos ver.

No es aconsejable desactivar permanentemente los informes de error ya que difié:oltala localización y corrección de los errores del código.

Desencadenar errores propios

La función trigger_error () se puede utilizar para desencadenar nuestros propios errores. Los errores creados de esta forma se procesan de la misma forma que los errores PHP convencionales.

La función requiere un mensaje de error y, opcionalmente,se le puede asignar un tipo de error. El tipo de error debe ser E_USBR_BRROR, E_USER_NARHING o E USBR NOTICB. Si no se especifica uno, el predeterminadoes B_USBR_ NOTICB.

- La forma de utilizar trigger _error ( ) es la siguiente:

```
trigger_errorl  'Thi•  coeputer  vill  ••lf  deatruct  in 15 •acODd•'.   B_USBR_lfAJUrlJIGI,
```

Solucionar errores con elegancia

Si tiene conocimientos de C++ o de Java, puede que eche de menos el procesamiento de excepciones cuando utilice PHP. Las excepciones permiten que las funciones indiquen que se ha producido un error y dejan la resolución del mismo a un cOAb'Glador aexcepdoMI Lu ucepdoMI apasec:ieroA e11 la veaión 5 de PHP y constituyen una excelente forma de solucionar errores en proyectos de gran tamaño. Como ya las analizamos en un capitulo anterior no volveremos a incluir su descripción.

Si desea que su código funcione con PHP4, puede siml;da~ un comportamiento parecido por medio de errores desencadenados por el usuano y controladores de error definidos por el usuario. Ya hemos visto cómo desencadenar errores personalizados. También puede proporcionar sus propios controladores para capturar errores.

Ya hemos visto cómo podemos desencadenar nuestros propios errores. También podemos utilizar nuestros propios controladores de error para capturar errores.

La función Ht error handler () nos permite proporcionar una función que se invocará cuando se produzcan errores en el nivel de usuario, advertencias y avisos. Podemos configurar set_error_handler () con el nombre de la función que queramos utilizar como controlador de errores.

Nuestra función de procesamiento de errores debe adoptar dos parámetros: un tipo de error y un mensaje de error. En función de estas dos variables, la función puede decidir cómo procesar el error. El tipo de error debe ser una de las constantes de tipo de error definidas. El mensaje de error es una cadena descriptiva.

Una llamada a ••t_error_handler () tiene este aspecto:

Despufs de indicar que utilice una función con el nombre myBrrorHandler (), debemos proporcionar una función con dicho nombre, que debe tener ei siguiente prototipo:

```
My_error_halldler(int  t1JK)_er.ror,  •tring  ~••i•:-er.ror
            [,  atringU"CIJ11Ī0_arror 1, intllllu_ error 1, arrayoaataxto~error)) J ))
```

pero cuyo funcionamiento depende de nosotros.

Los parámetros pasados a la función de controlador son los siguientes:

- El tipo de error
- El mensaje de enor
- El archivo en el que se ha producido el error
- La lfnea en la que se ha producido el error
- La tabla de sfmbolos, es decir, un conjunto de todas las variables y sus valores en el momento de producirse el error

Entre las acciones lógicas destacamos las siguientes:

- Mostrar el mensaje de error propon:ionado
- Almacenar información en un archivo de registro
- Enviar el error por correo electrónico a una dirección
- Pinalizar la secuencia de comandos con una llamada para salir

En el listado 25.2 se incluye una secuencia de comandos que declara un controlador de error, lo configura por medio de •et_error_handler () y, tras ello, genera algunos errores.

u.t.do 25.2. h•ndle.php. Estm ucuencIli de c:omllndos declara un controlador de enur psrsonellzado y genera distintose ——.

```
c?php
   11 La funci6n de control de error
   function l!ylrrorRandler (Serrno,  Serr•tr,  $errfile,  $errlinel
   (
      echo "cbr />ctable bgcolor•'lcccccc'>ctr>ctd>
            cp>cb>SRROR:c/b> Serretrc/p>
            cp>Pl•a••  try again,  or contact u•  and tell ue that
            thll error occurred in line $errline  of file  '$errfile•c/p>"¡
      if  (Serrno  ••  I_USSR_IIRllOR)
      (
         echo  'cp>Thia  error wae  fatal,  prograia endingc/p>'¡
         echo  'c/td>c/tr>c/table>'l
         //cierre  loe  recureoe abiertoe,  incluyendo  el  pie de p6gina,  etc.
         exitı
      }
      echo  'c/td>c/tr>c/tab~e>'I
```



```
)
// Defina el controlador de error
eet_error_handler('11Yllrrorllelldler•)1

//deeencadene dietintoe nivel•• de error
trigger_error('Trigger  functioa callad',  I_UBIR_II0'1'ICSI ı
topen( 'nofile',  •r•  11
trlgger_error(•'l'hi•  computer ia beige•,  1_us11_11A1111IIIG11
include  (nofile•)1
trlgger_error! ''l'hie computer will Hlf  deetruct in 15 ••conde•,  1_usn_lllll0JIl  ı
```

El resultado de esta sec:uenc:iade comandos se muestra en la figura 25.1.

Flgurm 25.1. SI utlllZllmos nuestn, propio controlador de errores podemos mostrar mensajes de error mM amistosos que los de PHP.

Este controlador de errores personalizadono hace nada mú aparte del comportamiento predeterminado. Como hemos escrito este código personalmente, puede hacer cualquier cosa. Nos permite elegir lo que los usuarios verúa cuando se produzca un error y cómo presentar esta informadón de forma que se ajuste al resto del sitio. Es mú, nos permite decidir lo que sucede: que prosiga la secuenda de comandos, que el mensajese registre o se muestre, o que se avise automáticamente a la asistendalftnlca.

EL_HOLISTICO@HOTMAIL.COM - 2006

A. continuación

En el siguiente capítulo sobre autenticación y personalización de usuarios, comenzaremos a generar nuestro primer proyecto. En éste, aprenderemos a reconocer usuarios que regresan a nuestro sitio y a modificar nuestros contenidos apropiadamente.

EL_HOLISTICO@HOTMAIL.COM - 2006

En este proyecto haremos que los usuarios se registren en nuestro sitio. Tras ello, podremos saber en quf est6n interesados y podremos mostrarles el contenido apropiado. Es lo que se denomina personalización de usuarios. Este proyecto en concreto permitirá a los usuarios generar una serie de marcadores en la Web así como sugerir otros enlaces que les pu.edan interesar en función de su comportamiento anterior.

De forma m6s general, la personalización de usuarios se puede utilizar prácticamente en cualquier aplicación Web para mostrar a los usuarios el contenido que quieren en el formato que desean.

En este proyecto, y en los posteriores, comenzaremos con una serie de requisitos similares a los que se reciben de un diente. Convertiremos estos requisitos en un conjunto de componentes de la solución, generaremos un diset\o para combinar dichos componentes y, tras ello, los implementaremos. En este proyecto, implementaremos la siguiente funcionalidad:

- Iniciar sesión y autenticar usuarios
- Administrar contraseñas
- Registrar las preferencias de los usuarios
- Personalizar contenidos
- Recomendar contenidos en función de la información existente sobre un usuario

EL_HOLISTICO@HOTMAIL.COM - 2006

El problema

Nuestro objetivo es generar un prototipo para un sistema de creación de marcadores en linea, que denominaremos PHPBookmar.ky que es similar (pero con menor funcionalidad) al disponible en Bac.kflip(http://backflip.com).

Nuestro sistema permitirá conectarse a los usuarios y almacenar sus marcadores personales así como obtener recomendaciones de otros sitios que puedan estar interesados en visitar en función de sus preferencias personales.

Estos requisitos de solución se agrupan en tres categorías principales.

- Es necesario poder identificar a los usuarios individualmente. Necesitamos algún método de autenticación.
- Necesitamos poder almacenar marcadores de un determinado usuario. Los usuarios deben tener la opción de añadir y eliminar marcadores.
- Necesitamos poder recomendar a los usuarios sitios en los que puedan estar interesados, en función de lo q~e conozcamos de los usuarios.

Componentes de la solución

U-na vez detetnunados I0S requisifós"cfelsiitema, podemos empezar a diseñar la solu!=lón y sus componentes. Veremos posibles soluciones para los tres requisitos printipales que acabamos de enumerar.

Identificar y personalizar usuarios

Existen distintas alternativas para autenticar a los usuarios, como ya hemos visto antes en el libro. Como queremos vincular a un usuario a cierta información personalizada, almacenaremos la conexión y la contraseña del mismo en una base de datos MySQL y realizaremos la autenticación en función de la base de datos.

Si vamos a permitir que los usuarios se registren con un nombre de usuario y una contraseña, necesitaremos los siguientes componentes:

- Los usuarios deben poder registrarse con un nombre de usuario y una contraseña. Será necesario restringir de algún modo la longitud y el formato de ~tos datos. Tendremos que almacenar las contraseñas en un formato codifi--éado por motivos de seguridad.
- Los usuarios deben poder iniciar sesión con los detalles que hayan proporcionado durante el proceso de registro.

- Los usuarios deben poder desconectarse cuando hayan terminado de utilizar el sitio. No es un atpecto muy importante si utilizan el sitio "desde su PC doméstico, pero sí, en cuanto a seguridad se refiere, si lo utilizan desde un PC compartido.
- El sitio debe poder comprobar si un usuario está conectado o no, y acceder a datos del usuario conectado.
- Los usuarios deben poder cambiar su contraseña como medida de seguridad.
- Los usuarios olvidarán sus contraseñas. Deben poder restablecerlas sin necesidad de que les ayudemos personalmente. Una forma habitual de hacerlo consiste en enviar al usuario la contraseña a la dirección de correo electrónico que haya indicado durante el registro. Esto significa que tendremos que almacenar su dirección de correo electrónico d~te el registro. Como almacenamos las contraseñas codificadas y no podemos descodificar la contraseña original, tendremos que generar una nueva, configurarla y enviársela al usuario.

Escribiremos funciones para todas estas opciones. La mayoría de ellas se podrán reutilizar, o al menos con mínimos cambios, en otros proyectos.

Almacenar marcadores

Para almacenar los marcadores de un usuario, necesitamos definir algún espacio en nuestra base de datos MySQL También necesitaremos la si~te funcionalidad:

- Los usuarios deben poder recuperar y ver sus marcadores.
- Los usuarios deben poder añadir nuevos marcadores. Debemos comprobar que se trate de URL válidos.
- Los usuarios deben poder eliminar marcadores.

Como dijimos antes, escribiremos funciones para cada una de estas opciones.

Recomendar marcadores

Podemos adoptar diferentes enfoques para recomendar marcadores a un usuario. Podríamos recomendar el más popular o los más populares de un determinado tema. Para este proyecto, implementaremos un sistema de sugerencias afines• que busque usuarios que tengan el mismo marcador que nuestro usuario conectado y sugieran sus marcadores a nuestro usuario. Para evitar la recomendación de marcadores personales, solamente recomendaremos marcadores almacenados por más de un usuario. De nuevo, podemos escribir una función para implementar esta funcionalidad.

EL_HOLISTICO@HOTMAIL.COM - 2006

Presentación de la solución

Después de realizar algunos bocetos, hemos obtenido el diagrama de flujo d~- sistema que mostramos en la figura 26.1.

Flgu,. 21.1. Este dlagnlmll muellre las posibles rutas del liltemll PHPBookmark.

Generaremos un módulo para cada cuadro de este diagrama; algunos necesitarán una secuencia de comandos y otros dos. Tambiln definiremos bibliotecas de funciones para:

- Autenticación de usuarios
- Almacenamiento y recuperación de marcadores
- Validación de datos
- Conexiones a bues de datos
- Representación en el navegador. Incluiremos todo el código HTML en esta biblioteca de funciones, asegumndonos de que la presentación visual es coherente a lo largo del sitio (se trata del enfoque de API de fundones para separar lógica y contenido).

También será necesario disei\ar una base de datos para el sistema.

Analizaremos la solución con mú detalle, pero todo el código de esta aplicación se incluye en el archivo correspondiente a este capítulo del CD-ROM. En la tabla 26.1 se recoge un resumen de los archivos incluidos.

En primer lugar implementaremos la base de datos MySQL para esta aplicación ya que la necesitaremos prácticamente con todas las opciones restantes para que funcionen.

Tablli 21.1. Ardllvoe de la apbc:16n PHP8ookmarlc.

bookllarka.aql	Instrucciones SQL pa,. crear la base da datos PHPBoolrmark
l09in.php	Piglna prlnclpal con el fannularlo de conexi6n al llalllma
r419iater_fora.php	FO(fflUlario para que laa u.... .. reglsnn .... llstema
r419iater_new.php	Secuencia de comandos para procesar nuevos rwglslroa
forgot_fora.php	Formularlo que deban campletar laa u.... ll han olvidado su connsefta
forgot_paaawd.php	Secuencia da comandos para reataurar contraseftas olvidadas
Mllber.php	PAglna principal de un usuarto, con una vista de todos 1111 marcadofn aclualN
add_bll_fora.php	Fonnularto para alladlr nuevos marcadores
add_llllla.php	Secuencia de comandos para alladlr loa nuevoa marca-dcns a la baH de dalD8
dalete_bu.php	Secuencia de comandos para allrnlnar marcadores ... lacclonadoS da la Hita dal u..io
reco.-nd. php	Secuencia de comandos para .ugerlr recomendaclonff a un usuario, en func:i6n de usuartoa con irllll'eMs limitares
change_paH'llld_fora.php	Formularlo que deben completar los mlembroa para cambiar 1111 contraseftaS
change_paaawd.php	Secuencia de comandos para cambiar la contraNfta de un usuario en la bale de datos
logout.php	Secuencia de comandos para desconectar a un usuarto de la apllcación
boot..rk_fna.php	Colacclón de archivos de lnclusl6n para la aplG8c:Mn
data_valid_fna.pbp	Funciones para valldar daa lntraducidos par el usuario
db_fna.php	Funclonea para CXIMCIllrN a la baH de daIN
uaer_auth_fna.php	FuncloneS para la aul9ntlcael6n da usuarios
url_fna. php	Funclonea para alladlr y allmlnar man:adcns , y para hacer recomendaclonas
output_fna.php	Funciones para aplicar formato HTML a los resultados
book..rk.gif	Logotipo de PHPBookm.ic

EL_HOLISTICO@HOTMAIL.COM - 2006

Tras ello, anallzaremoe el código en el mismo orden en el que se ha escrito, empezando desde la pigina inicial, pasando por la autenticación de usuarios, el almacenamiento y recuperación de marcadores y, por último, las recomendaciones. Este orden es bastante lógico; basta con definir las dependencias y generar en primer lugar los elementos que necesitaremos para los módulos posteriores.

Nota

Para que el código de este proyecto funcione como eati escrito, tendnl que activar las comlllaa mjgicaa. Si no lo ha hecho anteriormente, tenctnl que alladir addalHbH <> a loa datos que incluya en la base de datos y stripalaabe• <> en loa datos recibidos de la base de datos. Lo hemos utilizado como m6todo abreviado.

Implementar la base de datos

P.ara la base de datos PHPBookmark sólo necesitamos un esquema bastante sencillo. Tendremos que almacenar usuarios asf como sus direcciones de correo elecp-ónico y contrasems. Tambil!n será necesario almacenar el URL de un marcador. Un usuario puede tener ".arios marcadores y varios usuarios pueden registrar el mismo marcador. Por lo tanto, tendremos dos tablas, user y book-rk, como se indica en la figura 26.2.

21.2. ElqUel'IIB de la b8M de dlltoa pera el alatema PHPBoolunsltl.

La tabla uHr almacenará el nombre de usuario (que es la clave principal), la contr_,. y la dincci6n de correo elerctrónico. La tabla bookmark almacenará pares cl,e nombre de usuario y marcador (bm.)JRL). 8 nombre de usuario de esta tabla•hará referencia al nombre de usuario de la tabla uaer.

m SQL para aear esta base de datos~ y para crear 1m usuario para ralizar la conexión a la bue de datos desde la Web, se muestra en el listado 26.1. Deberla modificarlo si piensa utilizarlo en su sistema y cambiar la contraseila del usuario por algo mú seguro.

u.tado H.1. boolcmerka.lCII. Ardlivo SQL pera canftgurar la bue de dllloa de man:adOfaa.

```
create databa•• boolnllrke,
IIII• boolmarker

create table u..r   (
   ueenia119 ,rerehar(l,l priaery key,
   paeevd cbar(40I  not null,
   ... II YarcharllODI  not null
);

create table boolalark I
   u•eru- ,rarcharl1'l  not null,
   ba URL Yarcbar(2551  not IIIIII,
   illdez  (UIII'IWIMI),
   inclu:  (.i.  URLl
   pri-ry    key(uee~.  bta_llRLl
!!'

grant Nlect,  iuert,  update, delete
OII boolmarke.•
to  ~ ~ relocalhoet identified by  •paeevord•;
```

Püede configurar esta base de da.tos en su sistema si ejecuta este conjunto de comandos como el usuario raíz de MySQL Puedehacerlo con el siguiente comando en la línea de comandos de su sistema:

```
ayeql •u root •p e book-rke. eql
```

Tras ello, le instará a que introduzca su contrasefta. Una vez configurada la base de datos, implementaremos el sitio &ásico.

Implementar el sitio básico

La primera página que generaremos será login.php, que permite a los usuarios conectarse al sistema. El código de esta primen página se incluye en el listado 26.2.

Lisiado 21.2. logln.php. Piglna Iniciel del IiStelIla PHPBoolaJ!erk•

```
c?pbp
   require_oace(•bookmark_fae.php'I;
   do_htal_h..der(''I;

   dieplay_eite_info(I;
   dieplay_login_fon,();

   do_btlll_footerlll
?>
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Este código es muy sencillo. Búicamente invoca funciones del API de funcfones que construiremos para esta aplicación. Veremos los detalles de estas funciones en breve. Con tan sólo observar el an:hivo, comprobamos que se incluye otro (que contiene las funciones) y tras ello se invocan alpnas funciones para representar un encabezado HTML. mostrar contenidos y representar un pie de p4gina HTML

El resultado de 4:9ta secuencia de comandos se muestra en la figura 26.3.

FIgur11 21..3. La péglna lnicial del sistema PHPBookmark sa ganara por medio de las funciones de repreuntacl6n HTML de logln.php.

Todas las funciones del sistema se incluyen en el archivo bookmark_fn• . php, que mostramos en el listado 26.3.

Llatado 21.3. bookmark_fns.php. Archivo que lncluye lu funciones de la apllcacl6n de man:adonls.

```
c?pbp
   // Podn:,. incluir eete archivo en todae nueatroa arcbivoa
   11 de ••ta foraa, todo• loa arcbivoa contendr6n todaa nueatraa funcione•
   require_once(•data_valid_fna.pbp'l1
   require_once('cDl_fna.pbp'l;
   require_once1•uaer_autb_fna.pbp'l1
   require_once(•output_fna.pbp');
   require_once( 'url_fna.pbp• ll
?>
```


Como puede comprobar, el archivo es un contenedor de los otrol cinco archivos que utilizaremos en la aplicaddri. Lo hemos estructurado de esta forma ya que las funciones se agrupan en categorías lógicas. Algunas de estas categorías pueden resultarle útiles para otros proyectos, por lo que hemos incluido cada grupo de funciones en un an:hivo distinto, para que sepa dónde encontrarlas cuando las necesite. Hemos creado el archivo bookmark_fns.php porque utilizaremos la mayoría de los cinco an:hivos de funciones en nuestras secuencias de comandos. Resulta mú sencillo incluir este an:hivo en cada iecueilda cle comandos que tener cinco instrucciones require.

En este caso en concreto, utilizaremos funciones del archivo output_fns. php. Son funciones muy sencillas que dan como resultado HTML sencillo. Este archivo induyelascuatrofuncionesquehemosutilizadoenlogin.php,esdedr,do_html_header (), display_site_info(), dieplay_login_form() y do_htal_footer () entre otras.

No analizaremos todas estu funciones detalladamente sino que las veremos en un ejemplo. El listado 26.4 incluye el c6digo de do_html_header ().

Ustado 21.,. Funcl6n do_hmll_header() de oulput_fns.php. Esta funcl6n representa el encabezado esfjndar que apareceré en llodn In péglnH de la a~ .

```
fwaction do_btal_beader(Stitlel
(
   // iapriu un encabeaado BftlL

   cbtal>
   cbaad>
      ctitle>c?pbp ecbo $title;l>c/title>
      catyle>
         body { font·f..ily, Arial, Belvatica, aana·aerifl font·aiae, llpx /
         11, td { font·fallily, Arial, Belvatica, aana-eerifl font·aiae, llpx}
         br { color, l3333ccl vidtb-3001 text-align•left}
         a { color, 1000000}
      c/atyle>
   c/bead>
   cbody>
   cimg arc•"boolalark.gif• alt•"PBPbook..rk logo• bo:rder••0•
         align·•left• valign··bottoa• beigbta•ss• vidtb•°57"  ,,.
   cbl>ldll,ap;l'IIPboolalarkc/bl>
   cbr />
c?php
   ifCStitlel
      do_btal_baading($titlel;
```

Co~ puede apreciar, la única lógica de esta fundón consiste en aftadir el correspondiente título y encabezado a la p4gina. Las otras funciones que hemos utilizado en login.phpson similares. La funcidndisplay _eite_info (l aftade texto general sobre el sitio; display_login_formO muestra el formulario de color gris de la figura 26.3ydo_btml_footer <l aftadeun piedep4gina HTMLest4ndar alap4gina.

EL_HOLISTICO@HOTMAIL.COM - 2006

Lu ventajas de aislar o eliminar HTML del flujo de lógica nonnal se analizaron en un capitulo anterior. En este cuo utilizaremos el enfoque de API de funciones y, en el siguiente capitulo, el enfoque basado en plantillas, pua ver el contraste.

Si observa la figura 26.3 comprobará que en esta p4gina hay tres opciones: los usuarios pueden registrarse, conectarse si ya se han registrado o restablecer su contraseña si la han olvidado. Para implementar estos m6dulos, pasaremos al siguiente apartado, la autenlicaóón de usuarios.

Implementar la autenticación de usuarios

Existen cuatro elementos principales para el m6dulo de autenlicaóón de usuarios: registro de usuarios, conexión y desconexión, modificación de contraseftas y restablec:iiniento de contraseñas. Veremos cada uno de ellos con mayor detalle.

Registro

Pa.ra registrar a un usuario, necesitamos obtener sus detalles por medio de un formulario e introducirlos en la base de datos. Cuando el usuario pulse sobre el enlace Not a member? de la fJ'gina login.php, accederá a un formulario de registrogeneradoporregiater_form.php.Estasecuenciadecomandossemuestra en el listado 265.

Uiüiilo 213: régii(et.:_fcinn.php. este fonnuíarfo permite a I0S usu.io. registnirae en PHPBookmarks.

```
c?pbp
   require_once(•bookmark_faa.php');
   do_ht:al_h..der('U••r  Regi•t:r•t:ion•);

   diepl•y_regi•t:ret:ion_fore();

   do_ht:lll_footerCI
```

Como en el caso anterior, comprobará que es una página muy sencilla y que sólo invoca funciones de la biblioteca de resultados de output_fna. php. El resultado de esta secuencia de comandos se muestra en la figura 26.4.

El formulario de color gris de esta página se representa por medio de la función diaplay_regiatration_form (), incluida en output_fna. php. Cuando el usuario pulsa el botónRegister, accede a la secuencia de comandos regiater _nev. php, secuencia que se recoge en el listado 26.6.

Ual9do 28.1. reai.ter_new.pllp. Esta sec:uencia de comandol valida los nuevos datoa del usuario y los allade a la base da datos.

```
c?pbp •.
   // incluya archivo• da funcione• para e•t:• aplicaciOa
```



```
Crear un sitio
//cr.. nombre• de variable corto•
s...il•S_POS'rc•...i1•11
Sueeraaae•S_POST[•ueeraaae•J
$paeelld•S_P08'r('paeewd•J,
$paeelld2•$_POST[•paeellda•J
// inicie wur ..•iOa que puede neceeit:ar 116• tarde
// imciela ahora ya que daba ir aat:ee da loe encabesadoe
eeeeion_etart:[1 •
t:ry
(
 . 11 ca.pruebe que loe foraularioe eetAn completado•
  if (!filled_out:($_l'OSTII
  (
    t:hrow —   .Bzcept:ionC'You have aot: tillad   t:he fore out: correctly '
                        ''- pl...• go back ami t:ry again. '1;


  // direcciOa de.correo elect:r&lica no 9'1ida
  if 11valid_...i1cs...i111
  (
    t:hrow —   IIXA:ept:J.oaC"lliat: ie aot: a val.id —u addreee. P-1.... go back •
                        .• aad t:ry again.'1,


  // la• cont:ra•efta8 no coinciden
  if  !Spae•lld !• Spaeelld2!
  (
    t:brÓw new Bxcept:ion•i•The pa••word• you ... l:ered do II01: ..1:ch
                        '' • ple••• go bacll aad try a,aaln. '1;


  // compruebe que la 1~it:ud  de la• cont:reeetl.ae •• correcta
  if (et:rlea(Spa••lldlc,  ll et:rlenCSpaeelldl >1,!
  (
    th~  n- lzcept:icmC'Your paeeword — •t: be at: leaet:, charact:er• long.'
                        .'Plea .. 90 back aad t:ry again.'I;


  // ca.pruebe que la 10119it:ud del nombre de ueuerio ee correcta
  if (et:rlen(Sueeraa.el>1&!
  (
    t:hrow —  Bxcept:iaal'Your u••=-  ...l: be leH l:llan 17 cbaract:en long.'
                        .'Pleaee 90 back aad t:ry again.'I;


  // lnt:eat:e regiet:rar
  // eet:a fuaciOa t:allbiln puede 9eaerar una excepciOa
  regiet:er(Su•eraame, $...il, $pae811dl!
  11 regiet:re la ftriable de eeeiOa
  $_SBSSIO.(•vaUd_u ..r•J • Sueeraame,

  // ofresca un enlace a la plglla  de lliembro•
  do_ht:al_header('Regi•t:rat:ion eucceeaful'l;
  echo 'Your regiet:rat:ion wa• eucceeeful.Qo t:o t:he ...tier• PA9• '
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
        .•to  atart  aettiag up your boolcmarkal'1
    do_htal_url('aeaber.php•,   •ao to Maber•   page'l ı

    // finalice  la pagina
    do_htal_footerl),
)
eateh  (Sxeeption  Sel
(
    do htal  header1'Probl-, '1 ı
    eebo se:>getlleaaage(l1
    do htal  footer();
    –rt,  –

?>
```

, ... 21.4. El formul8rto de reglslro rec:uperll los detahs necesartos 1*8 18 base de dllfol. Loe uau•rlos deben .. c:rtblr 1u conlnlHII• dol WC81, por II se equlVOC11n.

Es la primera secuencia de comandos con cierta complejidad que hemos visto en esta aplicación. La secuencia comienza incluyendo los archivos de funciones de 1• aplicación e inicia una eesión (cuando el usuario está registrado, crearemos su nombre de usuario como variable de sesión, como hicimos en un c•pítulo anterior).

El cuerpo de la secuencia de comandos se concentra en un bloque try y• que comprobamos una 1erie de condiciones. Si alguna de ellas falla, la ejecución pasa •l bloque catch,. que veremos en breve. Tras ello, validamos los datos introduddos por el usuario. Existen varias condiciones que debemos probar:

- Comprobar que el formulario 1e ha completado. Para probarlo, invocamos la función filled_out () de esta forma:

  ```
  if  (lfilled_outlS_l'OdTII
  ```


Es una de las funciones qµe hemos merito ¡,enc,Mlmente. Se encuentra en la biblioteclde funciones del uchivodata_valid_fna .pbp. LaUlalfzuemos mú adelante.

- –Comprobar que la direa:idn de corno electr6nic:o proporcionada es válida. Lo probaremos de esta foriná:

  ```
  if  (valid_...il(S...illl
  ```

  De nuevo, es una de las funciones que hemos escrito, de la biblioteca data_va~ id_fna .pbp.

- Comprobar que las dos contraaeftas mgeridas por el usuario son idmticu, como se indica a continuación:

  ```
  if  l$puawd  I•  Spaand21
  ```

- Comprobar que el nombre de usuario y la contrasetla tienen la longitud correcta:

  ```
  if  (atrlen(Spa••wdlc,1
  ```

  y

  ```
  if  (atrlen(Suaern ...l>l'I
  ```

  En nuestro ejemplo, la contrasefta debe tener al menos 6 ca~$ para que resulte dificil descubrirla y menos de 17 para que se ajuste a fa bue de datos. La longitud máxima de la contrasefta no se restringe de esta forma, y• que se almacena en un hash SHAl, que siempre tendn 40 c:uacteres, independientemente de la longitud de la contrasefta.

Las funciones de valid•ción de datos que hemos utilizado, filled_out () y .valid_email C), se muestran en los listados 26.7 y 26.8 respectivamente.

Ll8INo 21.7. Func16n flllecl_out() de clm_v.ild_fna.php.Em funcl6n comprueba qUI el farmul8rlo N M camplemdo.

```
funetion  filled_out(Sfora_var•I
(
   // e0111pruebe que eada variable tiene un .alor
   foreaeh  (Sfora_var•  u  Skey -> Svalual
   (
       if  ( liHet  ISkeyl  ll  ($value  •  •  •'11
         return falae,

   return true;
```

U.tacto 21.8. Función v•ld_enmll() de data_vmld_fna.php.Em función comprueH 11 • dlrec:cl6n de corr90 electrónico •• rillda.

```
funetion valid_...il(SadcSreaal
(
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
// campnebe  que uaa direccida de correo e• v&lida
u (ereg( '' 1a-&A·ZO·t_\. \·) +e(a·SA·ZO·t\•l +\. (a·SA·ZO·t\·\. l+$', $addreH))
   return true;
elae
   raturn falae,
```

La función filled_out (l espera recibir UN matriz de variables que, por lo general, semn las matrices $_POSTo $_GET. Comprueba si estm completadas y devuelve true si lo estút y false en caso contrario.

La función valid ernail ‹ l utiliza las expresiones regulares que desarrollamos en uno de los primeros capítulos para validar direccioné de correo electrónico. Devuelve true si la dirección parece vallida y falH en caso contrario.

Una vez validados los datos introducidos, podemos intentar el registro del usuario. Si observa el listado 26.6, comprobm que se hace de esta forma:

```
regiater($uae~. s...11, $paaavdl;
// regiatra la variable de aHi6n
$_S&SSI011(velid_uaer•J • $uaarneme1

//ofresce un enlace• la pagina de aiallbroa
do btal beader('...iatration aucceaaful'I;
echo •Your regiatration waa aucceaaful.Go to tbe --.ra paga'
   ''to atart settin9 up your bookaarka!';
'do_btml_url( '-aier.pbp•, 'Go to ...t>er• pava• 1,

//: finalice la p&gina
dca;_,btai_roécerO /
```

CámO ve, invocamosla función register (1 con el noinbre de usuario, direcdón de correo electrónico y contruefta que hemos introducido. Si todo es correcto, registramos el nombre de usuario como variable de sesión y le proporcionamos al usuario un enlace a la p,gina principal de miembros. En la figura 26.S puede comprobar el resultado.

La función register (l se encuentra en la biblioteca incluida user _autb fns . pbp. Esta función se muestra en el listado 26.9.

Uamclo 21.t. Funci6n reglstar() de user_aulfl_lna.php. Eatll función intenm aft.clir la Informadón del nuevo uSWlrio a le bNe de d8lDs.

```
function regiacer($uaarnaaa, S... il, Spaaavordl
// regiacre UD nuevo uauario en la baae de datoa
// ...,_l•• true o UD — aja de error
(
   // c~t••e a la ba•• de dato•
   $CIIIIIII • db_connact(l,
      "
   // compruebe ai el a..bra de uauario •• axcluaivo
   Snllllll: • Sconn-..query(•aelecl: • fr- uaar vllara uaern-•'$uaarsume••11
::u CtSraaultl
      tbrow - Blllcepc1-c•c:a.a14 aoc .-cul:a query•t,
   if (Sranlt·>-_rowa>0I
```



```
   tbrow aev Bxception( 'Tbat -~ ia talren '
         .'• go back and cbooae uaotbar oae. 'I /

// ai ea correcto, iDclGyalo u la baH da datoa
$raault • Sconn->query(•inaart into uaar valuaa
                        ( '$uaarna.a'' abal(•$paa11VOrd.,, 'sa-il'!•, /
U (!$raauh:l
   tbrow aev bceptiOD('Could not regiatar you in databaaa '
                        ''- pluae try agaiD later. 'I /
raturn trua;
```

Figura 21.5. El registro ha sido satiafllctorio; el usuerio puede acceder a la pjgIna de miembros.

No hay nada nuevo en esta función, simplemente se conecta a la base de datos que definimoe anteriormente. Si el nomine de usuario seleccionado ya está utilizado o la base de datos no se puede actualizar, devolvem false. En caso contrario, sctualizam la base de datos y devolver, true. Debe saber que estamos realizando la conexión a la base de datos con una de las funciones que hemos escrito, db_connect (l. Esta función simplemente proporciona w,a ubicaáón que contiene el nombre de usuario y la contraseña para ainectarse a la base de datos. De esta forma, si cambiamos la contraseña de la base de datos, bastam con cambiar un archivo en nuestra aplicación. En el listado 26.10 podemos ver esta función.

Uatado 21.10. Función db_connec:t() de db_fna.php. Esta función se conecta' a ls bae de datos MySQL.

```
fUDction db_CODDect(1
(
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
Sreaull • naw ayaqli(•locallloat•, 'bll_uaar•, •paaaword', 'bookmarka'),
1f (1$rHult)
  tbrow new Sxcaption('Could not connect to databa•• aerver•),
•|••
return 'ireault;
```

Cuando los usuarios se registran, p~en conectane y desconectarse desde lu f>'ginas destinadas para ello, que generaremos a continuadón.

Iniciar sesión

Si los usuarios introducen sus datos en el formulario de login. php (vfase la figura 26.3) y lo envían, accedenn a la secuencia de ':°mandos member. php. Esta secuencia los conecta si han utilizado este formulario. Tambiffl muestra todos los marcadores relevantes para los usuarios que estm conectados. Es la parte central de toda la aplicación. En el listado 26.11 se incluye esta secuencia de comandos.

Umdo 21.11. member.php. Esta secuencia de comandos ea la parte prlnclpel de la apUcec16n.

```
c?pllp

// incluya loa arcllivoa de func16n para ••ta aplicaci6n
require once('boolmark fna.pllp');
••••ion::atart() '       -

//cree II0llbr•• da variable• corto•
Suaername • $ POST['uaerna..•J,
$paaawd • S_IIOST['paaawd'I,

if ($uaername ,, Spaaawdl
// acaban de intentar conectar••
(
   try
   (
      (login(Suaer-. $paaawd))
      // ai eatan en la baae de datoa, regiatre el Id. de uauario
      $_SUSl0ll['valid_uHr') • $uHrna.e;

   catcll
   (
      // conex16n no aatiafactoria
      do llt•l ll..dar('Probl ..:');
      eciio 'Y;u could not be logged in.
            You .uat be logg-4 in to view tilia page.';
      do_llt•l_url('login.pllp', 'Login'),
      do_lltal_footer();
      exlt;
```



```
check valid 11Hr[l1
11 obtenga ioa .ucadDrea qua llllya guardado ••ta ...ario
if (Surl_array • get_aaar_urla($_8SSSl0ll('valid_aaer•))l1
   diaplay_uaer_urla[Surl_arrayl,

// ofreaca UD ..ad da opcionea
cliaplay_uaar_.....cl;

do llt-1 footer(I,
?>-
```

Seguramente reconmca la lógica de esta sec:uenda. ya que estamos utilizando algunas de lu idNa de capltulos anteriores.

En primer lugar, comprobamos si el usuario proviene de la f>'gina principal, es dedr, si acabe de c:ompletar el formulario, e intentamae conectarlo de esta forma:

```
1f (Sunra.e " $paand)
// acaben ele intentar conactar••
(
   try
   (
      Uogin(Suaerna•, $paHwd))
      // ai ••tan en la baae de datoa, regiatre el ld. da .....a.
      $_818Sl0ll[',ralid_uaer•) • $aaerna..,
```

Comprobad que intentamos conectarlo por medio de una funci6n llamada login<), que hemos definido en la biblioteca user_auth_fns .php y cuyo código ana1Jzuemos en breve.

Si el usuario se COheCta satisfactoriamente, registramos su sesión como hicimos antes y ahnac:enamos el nombre de usuario en la variable de sesi4n valid_user.

Si todo funciona correctamente, le mostramos la P'gina de miembros:

```
do llt-1 INladar('S—'),
clleck_valid_uaer()1
// obtenga loa ..rcadorea que baya guardado Hte uaaario
if ($url_array • gat_uaer_urla($_818Sl0ll('valid_uaer•J))1
   diaplay_uaer_urla($url_array);

// ofreaca UD • lll6 da opcionea
diaplay_uaer_.enu();

do_llt•l_footer();
```

Esta f>'gina se ha creado con las funciones de resultados. Apreciar4 que hemos empleado algunas funciones nuevas como son check_valid_user(), de user_auth~fna.php;get_user_urla(),deurl_fns.phpydisplay_uaer_urla (), de output_fna.php. La fundón check_valid_user() comprueba si el usuario actual tiene una sesión registrada. Va dirigida a usuarioe que no se acaban de conectar, sino que estaln en el medio de una sesión. La función get_user_urls O obtiene los marcadores de un usuario de la base de datos y

EL_HOLISTICO@HOTMAIL.COM - 2006

display_user_urls () representa los marcadores en una tabla en el navegador. Analizaremos check_val id_user () en breve y las otras dos restantes en el apartado sobre almacenamiento y recuperadón de marcadores.

La secuencia de comandos member. pbp finaliza la página y muestra el menú con la función display_user_menu(). En la figura 26.6 se muestra un ejemplo del resultado que muestra member. pbp.

Flgu,. 21.i. La secuencia«-. comandol member.php comprueba que el usuario estj conectado, rec:upe,a y muean sus 11111rcectores y le ofrece un menú de opciones.

A continuación describiremos las funciones login () y cbeek_valid_user () con mayor detalle. La función login () se muestra en el listado 26.12.

Listado ~.12. Función login() de user_auth_fns.php. Esw funci6n comprueba los datoa de un USUMO en la base de elatos.

```
func:tion login(Su•arna-,  Spa••-rdl
// C:011Pruablt al nombra de uauario y la c:ontraaeila •n la ba•• d• dato•
// •t U d,  avu•lva  tru•
// •n c:aao c:ontrario g•nere una •xc-.,c:i6n
(
   //  c:oa6c:t•H  a la baaa da datoa
   $C:CIIUI • db_c:onaec:t();

   // c:ompru.blt ai al nombra da uauario •• u:c:luai-
   franlt • $c:onn-»qu•ry(•aelac:t • fr-  uaar
                          where 119eraame•Su••rna ..'
                          and pu•wd  • pa••-rd ‹'Spa••-rd'!'›,
   if  ℓtfranlt)
      tmow  naw lxc:eption('c—ld not log V-  ia.'11
   if !$r••ult-»aua_zow•»OI
      r•turn tru.;
   •|••
      thr-  a.w IIIXC:•Ptionl'C:0Uld not log you in.' I,
```

Como puede apredar, esta función se conecta a la base de datos y comprueba si existe un usuario con la combinación de nombre de usuario y contralei\a proporcionada. Devuelve true si lo hay o false si no lo hay o si no puede comprobar las credenciales del usuario.

La función cbeck_valid_uHr() no se conecta otra vez a la base de datos, sino que simplemente comprueba que el usuario tiene una sesión registrada, es decir, que ya se ha conectado. En el listado 26.13 podemos ver esta función.

Listado 21.13. Función check_valld_UMI'() de user_autti_fns.php. Es.. función comprueba si el usuario tiene una HIl6n vMlda.

```
func:tion c:hac:k_valid_uaarll
// c:ompru•blt aī al911ī•n •• ha CODac:tmdo y notifiqu..•lo •n c990  c:oetrario
(
   if li•••t1$_SIISSIOll[•valid_ua9r•J))
   {
      ac:II0 'Loggad  in aa '.$_SUSIOll('valid_ua•r'I.'.';
      ecbo •cbr / >;
   •|••
   {
      // DO ••t'n c:onec:tadoa
      do_ht•l_haading('Probl ..:');
      ac:ho •y-  ara not logg-4 in.cbr /»';
      do_html_:url1'log1a.php•,   'Login•1,
      do_html_foot•r<I,
      uīt;
```

Si el usuario no esti conectado, la función le indicará que debe conectarse pua ver esta página y le proporcionari un enlace a la pigina de conexión.

Cerrar sesión

Probablemente se habri fijado en el enlace Logout que aparece en el menú de la figura 26.6. Se trata de un enlace a la secuencia de comandos logout •pbp. El código de esta SICllellCia se muestra en el listado 26.14.

Usa.do 21.14. logout.php. Esw NCUendll de comandol flnalln la NIl6n de un usuario.

```
c?php

// inc:luya lo. arc:hivo• da func:i6n para •ata aplic:ac:i6n
raquira_onc:a('bookmark_fna.php'I;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
••••ion_atart lll
told_u•ar • t_SIIISSI011['valid_u•ar'J,
// almaeane para probar •i •••tallan• eonaet9do•
un•atlt_SB88I011)('valid_u•er•J,
tra•ult_de•t • ••••ion_deatroy(I;

// inida reaultado bt111
do_bt•l_baadar('Logging out•>,

U l luptyltold_uHr) 1
(
   (f ISreault_daat1

     // ai eataban eoneetadoa y ahora aatb de•eoneetado•
     eebo 'Logged out.cbr />'1
     do_btml_url('login.pbp•, 'Login'>,
   )
   alaa
   (
    // aataban eoneetadoa y no•• podlan de•eonaetar
     eebo 'Could DOt log you out.cbr />'I

)
aha
I
   // d no aataban eoneet9do• paro han aeeadido de algGA modo a ..ta pagina
   aebo 'You vera not loggad in, and ao bave not baan logged out.cbr />' 1
   do_btml_url('login.pbp', 'Login'11


do_btml_footer(I;
```

De nuevo, este código le resultar, familiar, ya que está basado en el código que diseftamos en un capítulo anterior.

Modificar contraseñas

Si un usuario utiliza la opción de menú Change Pasaword, accederá al formulario que mostramos en la figura 26.7. Este formulario se genera a partir de la secuencia de comandos change__paaawd_form.php. Se trata de una sencilla secuencia que utiliza las funciones de la biblioteca de resultados, por lo que no hemos incluido su código. Al enviar este formulario, se desencadena la secuencia de comandos change_paaawd. php, que mostramos en el listado 26.15.

Uatado 21.15. c:118nge_pauwd.php. Eatll secuencia de c:amanclos Intenta ct1mblar la contraHlla de un usuario.

```
c?pbp
   raquire_oneel'bookmark_fn•.pbp'11
   •a••lon_atart11;
   do_btml_beaderl'Cbanfia,i paaavord'1;

   // ene II011brea de Yariable eorto9
   told_paaallll • s_eo&T('old..,..allll'1,
   Snev_paaallll • S_l'0IIT(•nev_pa ..IIII•I,
   Snev_paaavd2 • t_l'0ST1•nev_paaalll2') ,

   t:ry
   (
      c:hac:k_valid_uaerCI,
      if 11fillad_out1S_l'0ST)1
         tbrov nev SZc:eptloal'You baft not Ullad out tbe fora e0111)letaly. '
                                  '' PlNH t:ry mgaiD. ') 1

      U ltnev_paaallll•Snev_paaavd21
      tbrov nev SXeaption1•Paa-rda entered vera not tbe .... llot ebanged. '>1
      if (atrlen(Snev_paaallll)c,1
         tbrov nev hieeption1'..., pmaavord —t be at laa•t , ebaraetar•.'
                                  •• T:ry agaln. ') 1

      // intenta aetuallsae16D
      eballge_pa••vord(S_SB88I011('valid_uaar•J, Sold_paaavd, Snev_paaavdl
      eebo 'Paa•vord ebanged. ';

   )
   eateh Cheeption Sal
   (
      eebo Se->gat11e••at•1)1
   )
   di•play_uaer_•nuCl,
   do_btlll_footerlll
?>
```

Figura 21.7. La secuencia de comandos change_puswd_form.php proporciona un fonnularlo desde el que los usuarios pueden ct1mblar aus contrasellas.

EL_HOLISTICO@HOTMAIL.COM - 2006

Esta secuencia de comandos comprueba si el usuario esd conectado (por medio de check_valid_user ()),que ha completado el formulario de contrasellas (por medio de filled_out ()) y que las nuevas contraseftas son idfnticas y tienen la longitud correcta. Nada nuevo. Si todo funciona correctunente, invoca la función change_paHwd() de esta forma:

```
cllaage_pa••vord(S_SSSSION[•valid_u•ar•J. Sold_pa••wd. Saew_pa••wdl:
acho •Pa••vord cbaaged. ":
```

Esta función es de la biblioteca uaer_auth_fna.php y su código lo incluimos en el listado 26.16.

Listado 21.11. Func:16n change_puawd() de user_aulh_rns.php. Esta runci6n intenta actualizar una contnisena de usuario en la base de datos.

```
function chan!J•_pa•-rd(S-rname, $old_pa••-rd, s-_pa••-rdl
// callbia la contraaalla da u••rna../old_pa••word por naw_pa••word
// davuelva trua o fal••
{
   // •i la contraaalla antigua aa corracta
   // cambia •ucontraaaila por naw_paa•vord y devuelva trua
   // an ca•o contrario, ganar• una axcapciOn
   109in(Su•arnama, Sold_pa••wordl,

   $CODII • db_cOIIIWCt(I:
   Sr••ult • $conn->q11ery( •updata u••r
                              ••t paaawd • abal('Snew pa-rd')
                              •b•r• u••rn&111e • 'Su••rn ... ••);
   U (t$ranltl
   t tJarow..naw Bxcaption('.tH•vord i:,wuld nqt ba c~gad. 'I :
   •]••
      raturn true, // ba cambiado corract...nta
```

La función comprueba que la antigua contraseila proporcionada es correcta, con ayuda de la fundón login<> que ya hemos descrito anteriormente. Si es conecta, la función realiza la conexión a la base de datos y actualiza la contraseña con el nuevo valor.

Restablecer contraseñas olvidadas

Ademú de modificar contraseilas, tendremos que hacer algo cuando el usuario olvide su contrasetla, algo muy habitual. Habral comprobado que en la página inic:ial. login. php, incluimos el enlace Forgolten your pasawon:t? para los usuarios con este problema. Mediante este enlace los usuarios acceden a la sec:uenda de comandos forgot_form_php, que utiliza las funciones de resultado para mostrar un formulario como el de la figura 26.8.

&ta sec:uenc:ia de comandos es muy sencilla; simplemente utiliza las funciones de resultado, por lo que no la desc:ribuemos detalladamente. Al enviar el formula-

rio, invoca la secuencia de comandos forgot_paaawd. php, que es mú interesante. En el listado 26.17podemos ver esta secuencia.

Figura 21.1. La secuencia de comandos forgot_form.php proporciona un formulario con el que los usuarios pueden solicitar el restablecimiento y el envio de sus contrasellas.

Listado ff. U: forgot.J)ailawd.php Este secuenda de wmandt11 1alablece ta contrasella de un usuario por veto, aleatorio y le envia la nueva.

```
c?pbp
   raquira onca("bool1:urk fn•.pbp°I;
   do_html:haadar(•Ra•atting pa••word•I:

   //craaci6n da un IIC11br• da variable corto
   Su•arname • S_,OST['u•arname•J;

   try
   {
      $pa•sword•r•••t_pa••word(Su•arna11al:
      notify_paa•word(Su•arna... $pa••word1;
      aebo 'Your naw pa••word ba• baan •ent to your —u addr•••.';

   catch
   (
      acbo 'YOII% pa•avord could not be n••t • pl•a•• try &gain latar.•;
   }
   do_btal_url('login.pbp•, 'Login'I:
   do_btal_footar(I;
?>
```

Como puede ap~r, esta secuencia de comandos utiliza dosfundones principales para realizar esta tarea: reaet_paasword () y notify_password () . Vea-

EL_HOLISTICO@HOTMAIL.COM - 2006

mos cada una de ellas. La función reae,t_paasword O genera una contrasefta aleatoria para el usuario y la ailade a la base de datos. Su c6digo se incluye en el listado 26.18.

Llatlldo21.11.Func:16nraet_penwonl()deuaer_aulh_fna.php.E•Nc:uenc1ac1ec:omandoe restablecela contraHfta ~· un uau.rlo con un valor aleatorio y le envla la nueva.

```
function reaet_paaaword($ua•r1M1..)
// eatabl•sca la contr..•fta del nombre de uauario con un valor al•atorio
// devuelva la nueva contr••efta o falee en caao d• fallo
(
   // obtenga una palabra aleatoria del diccionario con una longitud de,
   // a 13 caract•r••
   $new_paeeword • get_randoll_word(6, 13);

   if($new_J1aaa1lard• •falael
      throw n•w IXception('Could not generate new p..eword. '1 |
   // aftadale un na.ero C011Pr•ndido •ntre o y 999
   // para ..jorar liger•..nte la contraaella
   arand ((doubl•I •icroti..(I • 100000011
   $rand number • rand(0, 999);
   $n•w..iiaaaword '' $rand_nUllb•r;

   // defina con ••to la contraaefta del uauario en la baa• de datoe o
   /I devuelva falae
   $conn • db conn•ct(1,
  Sreault • iconn·>query( •update uaer
                          ••t paaawd • •bal('$new_paaaword')
                          wbare ueer!MI.. • '$ueerna..••1,
   if l1$rHultl
      tbrow new Sxception('Could not cbange paHword. '1; // no H ha II0dlficado
   •]••
      r•turn $new_paaeword1 // ae ha II0dificado correct...nt•
```

La función genera una contruefta aleatoria. Para ello obtiene una palabra aleatoria de un diccionario, utiliza la función get_random_wow () y le ailade como sufijo un nwnero aleatorio comprendido entre 0 y 999. La función get_random_word (1 se encuentra también en la biblioteca user_auth_fna .php y suc6digo se incluye en el listado 26.19.

Uatado 21.11. Funciónget_ranctom_word()de UNr_eulh_fna.php.E• func:Mn obtiene una palabra aleatoriade un diccionario y la utiliza para generar conlraHllaa.

```
function get_random_word($a1n_length, $max_l•ngth)
1/ obtenga del diccionario una palabra aleatoria comprendida entre la• doa
/I longitudea y devuflvala
{
    // gen•re una palabra aleatoria
   $word • '',
   !! no olvide callbiar eata ruta para qu• H ajuate a au •i•t•..
   $d1ct1onary • '/uer/dict/wrda•; // el diccionario iapell
   $fp • fopenl$dict1onary, •r•l,
   if (1$fpl
      r•turn fale•,
   $•1•• • flleaise($df.ctionaryl1

   // vaya a an punto al•atorio del diccionario
   erand ((doublel llicroti•O • 100000011
   $rand locatloa • rand(O, S•1••11
   f••ttl1<Sfp, $rand_locatlonl1

   // obtenga la algulente palabra de la longitud correcta en •l archivo
   whil• (atrlen($11ardlc $11.ln_length ||
      atrl•nl$110rdl>s-mt_l•ngtb || atratr($word, •••11

       if (feof($fpl!
         fa..k($fp, 011// ai ••ta al final, vaya beata •l principio
       $word • fgeta(Sfp, 101,
   // evlt• la pri..ra palabra ya que podr1a ••r parcial
       $word • fgeta($fp, 1011 // la contraa•fta potencial
   }i
   $110rd•tria($11ardl1 // recorte el \n final de fg•ta
   return $11ard;
```

Para que funcione, la función necesita un dicdonario. Si utiliza u&'listema Unix, el revisor ortogr6fico incorporado, ispell, contiene un diccionario ci!Jalabras, que normalmente se encuentra en /usr/dict/words como en este e--. o en /usr/share/dict/worda. Si no lo encuentra en ninguna de estas ubiéaciones, en la mayorfa de los sistemas lo podr6 localizar si escribe

```
locat• dict/ll0rda
```

Si utiliza algún otro sistema o no quiere instalar ispell, no se preocupe. Puede descargar listas de palabras como las utilizadas por ispell de la siguiente dirección: http://wordlist.sourcefoxge.net.

Este sitio también cuenta con diccionarios en muchos otros idiomas, por lo que si quiere utilizar una palabra aleatoria en noruego o esperanto, por e;entplo, puede descargar cualquiera de estos diccionarios. El formato de los archivos muestra cada palabra en una línea y las lfneas separadu entre sí.

Para obtener una palabra aleatoria de este archivo, seleccionamos una ubicadón aleatoria comprendida entre 0 y el tamafto del archivo, y leemos desde ese punto del archivo. Si leemos desde la ubicación aleatoria hasta la sigwente línea, probablemente obtendremos una palabra parcial, por lo que ignoramos la línea en la que hemos abierto el archivo y tomamos la siguiente palabra invocando dos veces fgeta (). La función tiene dos partes muy interesantes. Por un lado, si llegamos al final del archivo mientras buscamos una palabra, volvemos de nuevo al comienzo:

```
u (f-f($fp))
  faeekl$fp, 01l   // ai eata al final, vaya beata •l principio
```

Por otra parte, podemos buscar una palabra de una longitud determinada; comprobamos cada palabra que obtenemos del diccionario y, si no est4 cmnprendida

EL_HOLISTICO@HOTMAIL.COM - 2006

entre $ain_length y $max_lengbt, seguimos buscando. Almismo tiempo, también podemos eliminar palabrp con apóstrofes (comillas simples). Podríamos suprimirlos al utilizar la palabra, pero resulta más sencillo obtener la palabra exacta.

Volvamos a re•et_pa••word (). Una vez generada la nueva contraseila, actualizamos la base de datos para indicarlo y devolvemos la nueva contraseila a la secuencia de comandos principal. Esto se pasa a notify_paaaword(), que la enviar6 por correo electrónico al usuario. La función notify _paasword () se describe en el listado 26.20.

Listado 21.20. Funci6n nollfy_pauword() de user_aulh_fna.php. Esta funcl6n envla la connsefta restablecida por correo electrónico al uabarto.

```
function notify_paaaword(Suaerna... Spaaaword!
// iaforaie al U8uario que••  lla modificado au contraaeila
(
   $conn • di» connect|J:
   $reault • icoaa·>quary!"••lect   ...11 from uaer
                            wbere u••rna ... 'SU ..rDallNI'.,:
   if (ISrHultl
   (
     ;.tbrow aew Bxception(•C:ould aot fiad ...11 addr•••. 'I ,
   }
   •l•• if CSr.ault·>D'WII_row•• •01
   (
     throw ae• lxception(•c:ould not ,Und '
                            .' •–11 addr•••. 'J ;
                            // el IIOlllbre de uauaria. no ..U  en la bue de ütoe

   •]••
   (. .---- ..-
     srow • $reault·>fetcb_object0J;
     sa..11 • srow->...il:
     Sfrom • ·Vr—, aupportepbpbookmark \r\n•,
     s–•9 • ºYour PRPbook..rk pa••word ha• been cbanged to Spa••word \r\a•
       .•Pl•••• change it aext ti.. you logia.  \r\a•;

     it  C.. 111s... 11,  'PHPbookmark login intol:9ation•,  S • •g, Strom>›
       retu.ra true;
     •]••
       throw aew Bxceptioa|C:ould not ••ad ... 11.•1,
```

En es~ función, con un nombre de usuario y una contraseila basta con buscar la dirección de correo electrónico de dicho usuario en la base de datos y utilizar la ñinci6n —u ‹) de PHP para enviarla.

Seria más seguro proporcionar a los usuarios una verdadera contraseila aleatoria, ~rada a partir de una combinación de letras minúsculas y mayúsculas, números Y signos de puntuación, en lugar de nuestra palabra y número aleatorio. Sin embargo, una con~ como zigazg48 7 resulta más sencilla de leer y de escribir que una verdaderamente aleatoria. A menudo a los usuarios les resulta diffcil saber si

un c:arider de una cadena aleatoria es 0 u O (cero u O mayúscula) o 1 o l (el *uno* o la ele mindac:ula).

En nuestro sistema, el archivo de diccionario contiene apn>ximadamente 45.000 palabras. Si un pirata informático supiera cómo creamos las contraseilas y conociera el nombre de un usuario, tendria que probar con una media de 22.500.000 contraseilas para adivinar una. Este nivel de seguridad parece adecuado para este tipo de aplicación incluso si nuestros usuarios rechazan el consejo enviado por correo electrónico para que cambien su contraseila.

Implementar el almacenamiento y la recuperación de marcadores

A continuación veremos cómo se almacenan, recuperan y eliminan los marcadores de un usuario.

Añadir marcadores

Los usuarios pueden ailadir marcadores si pulsan el enlace Add BM del menll de usuario. De esta forma acceden al formulario mostrado en la figura 26.9.

Figura 21.9. La secuencia de comandos add_bm_fonn.php prapordona un fonnulario deede el que los usuarios pueden atladlr marcadores a sus páginas de mlll'l:9dor8e.

Como en casos anteriores, esta secuencia de comandos u~ múcamente las funciones de resultados, por lo que no la analizaremos en este apartado. Al enviar

EL_HOLISTICO@HOTMAIL.COM - 2006

el formulario> invoca la secuencia de comandos add_bma. php, que mostramos en el listado 26.21.

Llstaclo~l.2kadd_bms.php. Esta secuencia ele-comandos allade nuevos marcadores a la P'glna pe,90nal de un uau.io.

```
c?pbp
 require_once(•bookllark_fne.pbp')i
 eeeeion_etart])1

   //cree un n0llbre de variable corto
   $nev_url • $_1i'0ST[inev_url'l1

   do_btml_beader('Adding  bookllarke.)1

   try
   (
      check_valid ueer();
      if  (!filled_out!$_l'0ST))
      (
         tbrov new Bxception('Fora  not coapletely filled out.•),
      }
      Il compruebe el foinieto del URL
      if  letretr($new_url,  'bttp://')•  • •falee)
         $new_url • 'bttp://i.$new_url1

      // compruebe ei el URL ee valido
      if  (efopen($new_url,  •r•))
         tbrov new Sxception(••ot  a valid URL.')1
      // intente alladir un marcador
      add b•l$new_urll;
      ecbo  •aookmark added.'1

      // obtenga loe aarcadoree almacenado• por eete ueuario
      if  ($url_array•  get_ueer_urlel$_SIISSIO•C'valid_ueer•)));
         dieplay_ueer_urle!$url_array);
   }
   catcb  (Bxception  $e)
   (
      ecbo  $e•>getMeeeagell;
   }
   dieplay_ueer_..null,
   do_btal_footerll;
```

De nuevo, esta secuencia de comandos utiliza el modelo de validación, entradas de base de datos y resultados. Para validar, primero comprobamos si el usuario ha completado el formulario por medio de filled_out ().

Tru ellos, realizamos dos comprobaciones de URL. Primero, por medio de atrstr ( ), comprobamos si el URL comienza por ht tp: / /. Si no lo hace, lo aftadimos al inicio del URL Tras ello, podemos comprobar que el URL existe realmente. Como recordará de capftulos anteriores, podemos utilizar fopen <) para abrir un URL que en'lpiece por ht tp : //. Si podemos abrir este archivo, uumimos que el

URLes v61idoe invocamos la ~c:i6n add_bll () para aftadirlo a la base de datos. Esta función y el resto de funciones reladonadu con marcadores se encuentran en la biblioteca de funciones url_fns. php. En el listado 26.22 incluimos el c6digo de la funci6n add_bm( ).

u.tallo 21.22. Funcl6n add_bm() de Ull_fna.php. Esta funcl6n allade 11U1M1S marcadcnl a la bale de daloa.

```
fUDCtiaa add_bm($new_urll
(
   // Allada el nuevo marcador a la baee de datoe

   ecbo •Att111111ting to add •.bt•lepecialcbare($new_url).'cbr   />'1
   $valid_ueer • $_SIISSIO11[•valid_ueer•];

   $conn • db_connect(l1

   // incluya el nuevo Mrcador
   if  (1$conn->query(  •ineert  into bookllark valuee
                          ('$valid_uear•,   '$new_ur1'1')1
      tbrov new IIXCeption('aoolcmark could not be ineerted.'1 ! .....,..
                                                                        ''''_
   ret11m true;
```

Esta función es butante sencilla. Comprueba que un usuario no 1enga ya este man:ador enumerado en la base de datos (aunque es improbable que los usuarios introduzcan un marcador dos veces, es muy posible que quieran actualizar la p4gina). Si el marcador es nuevo, se introducirá en la base de datos.

Si se fija en add_bm.php, comprobará que lo .Utimo que hace es invocar get_user_urls () y display_uaer_urla (), lo mismo que member.php. A continuación analizaremos todas estas funciones.

Mostrar marcadores

En la secuencia de comandos member. php y en la función add_bm () utilizamos lu funciones get_uaer_urla () ydisplay_user_urls (). Estas funciones obtienen los marcadores del usuario de la base de datos y los muestran, respectivamente. La func:iónget uaer urla (> IN! encuentra en la biblioteca url_fns. php y lafunciónd iaplayjaaer.)na.php en la biblioteca output_fna .php.

En el listado 26.23encontrará la función get_uaer_urla ().

U.lado 21.23. Funcl6n get_user_urll() de urt_fna.php. Esta funcl6n recupera de la base de datos los marcadOrea de un usuario.

```
function get_ueer_urle($uearnaae!
(
   //extraiga de la baee de dato• todo• loe URL que baya almacanad4? eete ueurio
   $conn • db_connact();
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
Sre9111t • Sccma-~z;y(  ••elect t._UIIL
                        froa bookMrk
                        wbere U8eruaM  •  'Su•ernaae••)1
U  (!Sreeultl
   return false,

//cree 1111a Mtria de lo• UIIL
Surl_array • array();
for Tscount • 1;  Srow • Sr••ult·>fetch_row(); ++Scount)
|
   Surl_array[Scountl  • Srow[0];
|
return Surl_array;
```

Analicemos brevemente esta función. Adopta un nombre de usuario como parámetro y recupera los marcadores de dicho usuario de la base de datos. Devuelve una matriz de dichos URL o el valor false si los marcadores no se pueden recuperar. La matriz de get_user_urls O se puede pasar a display _user_urls (). Se trata, de nuevo, de una sencilla función de resultado HTML para imprimir los URL del usuario en formato de tabla, por lo que no la describiremos. En la figura 26.6 puede comprobar el upecto del resultado. La función se encarga de aftadir los URL a un formulario. Junto a cada URL aparece una casilla de verifica-~ón que permite seleccionar los marcadores para su eliminación.

Eliminar marcadores

C~o un usuario selecciona un marcador para su eliminación y pulsa la opción Delete BM del menú, se envía el formulario que contiene los URL. Cada una de las casillas de verificación se genera a partir del siguiente código de la función display_user_urls():

```
•cho •ctd>cinput type•\"checkbox\• n...•\"del_me[J\•
        value•\"$url\"></td>"1
```

El nombre de cada entrada es del_me CJ' Esto significa que, en la secuencia de comandos PHP activada por este formulario, tenmemos acceso a una matriz denominada $del_me que contiene todos los marcadores que se van a eliminar.

AJ pulsar sobre la opción Oelete BM se activa la secuencia de comandos delete_bms. php. Esta secuencia de comandos se muestra en el listado 26.24.

LI9tIIdo 21.24. delete_bms.php. Esta secuencia de comandos elimina msrcadonls de la baae de datos.

```
c?pllp
   require_once('booaark_fn•.pJlp •) ;
   ••••i~n-•tart();

   //cree DOllbre• de variable• cortos
   Sdel_M  •  $11TTP_llOST_VIUUl('del_me•J,
```



```
   do_htllll_header('Deleting bookmarka•);
   check valid u••r() ;
   U  (lfilled-out (SIITTP_l'08T_VARS))
   |
      echo 'You bave not cho••n any bookMrk• to delete.
         Pleaae try again.•;
      display uaer aenu(),
      do_ht•l:footer();
      exit;

   elae
   (
      if (count(Sdel_ ..,  >0I
      (
         foreach(Sdel_me u  Surl)
         |
            if ldelete_baCfvalid_uaer,  $Urlll
               echo 'Deleted  '.htmlapecialcbara(Surl).'.cbr     />';
            el••
               echo 'could not delate  '.htllllapeciflcharalSurl). '.cbr />'';


      .1 ••
         echo 'No bookMrk• aelected for deletioa•,
   }
   II obtenga loa aarcador•• que haya guardado ••t• usuario
   if (Surl~array • get_u..r_urla(Svalid_uaer)I;
      diaplay~u••r_urla(Surl_array);

   diaplay_uaer_menull,
   do_html_footer();
?>
```

La secuencia comienza con la ejecución de las validaciones habituales. Cuando sabemos que el usuario ha seleccionado algunos marcadores para su eliminación, los borramos con el siguiente bucle:

```
foreach(Sdel_me as surl)
|
   if ldelete t.Csvalid uaer, surlll
      echo 'Deleted '.ht:8lepecialchara($url).'.cbr      />';
   el••
      echo 'could not delete  '.htalapecialchar•(Surl).'.cbr    />',
```

Como puede comprobar, la fundón delete_bmO se encarga de eliminar los marcadores de la base de datos. Esta función se muestra en el listado 26.25.

Lls.. do 21.25. Funcl6n clelate_bm() de urt_fnl.php. ElfIl función almlna un solo msn:ador de la bese de dalos.

```
functioo delate_~(Su•er,  Surll
(
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
// borre un URL de la baae de dato•
$conn • db connect();
// borre el marcador
if (1$conn->query(     •dalete   froa  bookllark
                        where  uaem — •  '$uaft •  and  ba..:,ug••'$url''l)
   thr-  new sxcept ion('Boolmark could not be deletad '):
return true;
```

Como puede apreciar, se trata de nuevo de una sencilla función. Intenta eliminar el marcador de un determinado usuario de la base de datos. Pfjese en que queremos eliminar un par nombre de usuario-marcador concreto. Puede que otros usuarios tengan este URL marcado.

En la figura 26.10 puede ver un ejemplo de ejecución de la secuencia de comandos de eliminadón en nuestro sistema.

Flgu,. 21.10. La sec:uencla de comandos de ellmlnaci6n notlftca II usuario los marcadores borrados'/, tras elo, IIIUNtra los man:adcns restantes.

Al igual que en la secuencia de comandos add_bma . php, una vez realizados los cambios en la base de datos, mostramos la nueva lista de marcadores por medio de get_uaer_urla () y diaplay_uaer_urla (l.

Implementar recomendaciones

Por último, llegamos a la secuencia de comandos del enlace de recomendaciones, recommend. php. Existen qistintos enfoques a la hora de realizar recomendadones. Hemos optado por lo que d~amoe recomendacidn de "afinidades". Es dedr, buscaremos otros usuarios que tengan al menos un marcador similar al de nuestro usuario. Los marcadores restu:ttes de otros usuarios también le pueden interesar.

La forma m'8 sendlla de implementarlo como consulta SQL consiste en utilizar una subconsulta. En primer lugar ejecutamos la siguiente a,nsulta:

```
aelect diatinct(b2.u ........l
        fra. boolmark bl, boolmark b2
        where bl.uae-•'$Yalid     unr'
        and bl.uaeniamm  I• b2.uaern&1N1
        and bl.llll_llllL • b2.IIII_IIIIL
```

Esta consulta utiliza alias pan unir el marcador de la bue de datos a sf mismo, un concepto extru\o pero en ocasiones útil. Imagine que existen dos tablu de marcadores, una con el nombre bl y otra llamada b2. En bl, trabajamos con el usuario actual y sus marca——. En la otra tabla, analizamos loa marcadores de todos los usuarios restantes. Buscamos otros usuarios (b2. uaername) que tengan un URL similar al del usuario actual (bl •bm_tJRL • b2 . bll_tJRL) y que no sean el usuario actual {bl. uaername I• b2. username).

Esta consulta genera una lista de usuarios afines al nuestro. Por medio de esta lista, podemos buscar sus marcadores restantes con ayuda de la siguiente consulta:

```
aelect IIII_URL
fra. boolmark
where ueernam  in
        laelect diatinc:t:(b2.u•emame)
           froa boolalark bl, boolmerk b2
        where bl.uaername•'$Yalid_uaer•
           and bl.u•ema• l• b2. uaemame
           and bl,IIII_UIL • b2.IIII_IIIIL)
```

Aftadimos una aegunda subconsulta para filtrar los marcadores del usuario actual; si ya tiene un marcador, no tiene sentido rec:omencUnelo. Por último, aftadimoa un filtro por medio de la variable $popularity ya queno queremos recomendar ningún URL que sea demasiado personal, por lo que úniaunente sugeliremoa direcciones URL que un determinado número de loa otros usuarios de la lista de afines hayan marcado. La consulta definitiva tendd este aspecto:

```
aelect ba_ORL
fr:om boolmark
where uaerna•  in
        (••lect diatinct(b2.u•ern&11el
        fra. boolmark bl, boolmark b2
     where bl.uaernaM•'$Yalid_uaer'
        and bl.uaemaiae  I• b2,IIIIerna ..
        and bl.IIII_URL • b2,IIII_IIIIL)
and IIII URL not in
     -  leelect IIII URL
        froa boolmark
        where uaerna • •'$Yalid_uaer•J
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
group by i._url
bavi119 COUDtltim_url)>Spopularity
```

Si pensamos que nuestro sistema tendri un elevado número de usuarios, podrfamos ajustar Spopularity para que sólo sugiera direcciones URL que hayan marcado un elevado número de usuarios. Los URL marcados por una gran cantidad de usuarios seguramente sean de gran calidad y sean mú atractivos que una p4gina Web media.

La secuencia de comandos completa para realizar recomendaciones se describe en los listados 26.26 y 26.27. La secuencia de comandos principal se denomina recommend. pbp (vfase el listado 26.26). Invoca la funci6n de recomendación reco111111end_url• () de url_fna .php (vúseel listado 26.27).

Umdo 21.21.recommend.php. Esta secuanm de comandos sugiere algunos rmn:adores que le pueden Interesar a un usuario.

```
c?pllp
   raquira_once('bookllark_f118.pbp');
  ••••ion atart();
  ilo btl11-beader( 'a.c —— adiag wi.. •1;
  itry -
  (
     cbeck_valid_uaer(11
     $urla.  rec—nd urla1$_SRSSION(•valid_uaer'111
     diaplay~rec0111111endad_urla1$url•11
  |
   catcb(Bxception $el
   (
     acbo $e·>getlleaaagall;
   }
   diaplay_u•er_..nul);
   do_btal_footerl1;
```

Uatado 21.27. Función recommend_urls() de urt_fna.php. Esta secuencia de c:omandol reellza las rec:omandacione1.

```
function rec011199DC1_url•1$valid_uaar, $popularity • 11
{
   // Ofrecaao• a loa u•uario• recoaandacioaea a.-intaligantaa
   // Si ti•-  wa UJtL en ca.Gn con otro•, puede que ••t•n interaaadoa en
   // otroa ORL que 1•• parescan atractivoa
   Sconn • db C011Dect111
   // buaque ;tro•  usuario• con un URL aiailar al auyo
   //  coIIo ••ncillo _.todo para excluir pigin.. privada• da uauarioa y
   //aumentarla poaibilidad de recoaandar URL atractivo•
   // aapecificamoll UD nivel da popularidad a1Dimo
   // •i $popalarity • 1, —• da UD uauario debe tener
   // UD ORL ante• da que~· recoaandarlo
   tquary,• ₀Hlect bm_URL
         fraa  11oolmark
         wbera uaarnama in
```



```
         laalect diatinctlb2.uaar11m1a1
            fr— bookaark b1, bookaark b2
         wbare b1.uaarna.a•'$valid u..r'
            aad b1.u•arna.a 1• b2. ii'aarna.a
            and b1.ba .URL • b2.ba URL)
      aad ba_URL DOt 1n          -

            IHlect ba URL
            froa bookurk
            wbera ua•rna — •'$valid uaar'J
      group by bGI url              -
      baviag COUDt1ba_urll>$popularity•1

if  II ($raault • $CODD•>q11ary|$quary111
   tbrow naw lxcaption('Couldaot fiad any bookaarka to rae —— ad.•);
if 1$reault•>DUII rowa• •01
   tbrow naw lxceptical'Could not fiad any bookaark•  to rec-..cl.•J 1

$url• • array1);
10 cr.. una ..tria da loa IJIIL iaportantaa
1r 1$couat•01 $row • $raault·>fatcb_objactl1; Scouat++)

   $url•1$couatJ • $row•>ba_1JIIL;


raturn $urla1
```

En la figura 26.11 puede ver un ejemplodel resultadogenerado porrec:cmnend. php.

Flgun 21.11. La 1acuencla de comandol ha recomendado qm a nte u1u.rto le puede interaaaramazon.com. Otros dos usuarios de la bale de daioa tambi6n han marc:NOate sitio.

EL_HOLISTICO@HOTMAIL.COM - 2006

-lllllllll--c-:-:-:-:-:-:-:~-·-~-"-y-"po-~-"-•·-:-:--c-io n es------

Huta ahora tae.. mostnclo la ..,..... Weica de la apllcacWn JIIIPloabmk. Tam11Mna1111111clllltbata11,cl1mlawpa....,en11ei.919... c:amm i. ... .

- Un..._.•.Apepramlil ..... puai.NC 1--t-=--
- **IINlmrd,,,.,,,,_..... .,.um... ,.,.....•la.. c1ec1a1a1om ;**
- u• .....,. ..... ,..Wryedadllillbawrimytm1a1
- T~ pua que m muaidons acammdadol - _.. inlleligadal y
- Compi....ct6nckerrm. adidanal ..... ~ .....

..... - ..... fomact.apnnder

A continuación

lnelrC& fst1i1¡"tlü• e r•--•1a——quepmailir6ab usuarioe ..... ,. ....,,..... PJlfllll ane.de cmfinnar- pecli- doey ndm• ,.....,..,,.

27

de la compra

En este capítulo aprenderemos a diseñar un carro de la compra básico, que añadiremos a la base de datos Book-0-Rama que implementamos en capítulos anteriores.

También exploraremos otra opción: la configuración y el uso de un carro de la compra PHP de código abierto existente.

En caso de que no lo haya escuchado antes, el tfrmino carro de la compra (a veces también se denomina cesta de la compra) se utiliza para describir un mecanismo de compra en línea especffico.

Al navegar por nuestro catálogo en lfnea, puede atladir artículos a su carro de la compra. Cuando termine, se pasa por la caja de la tienda, es decir, se adquieren los artículos del carro.

Para poder implementar el carro de la compra, tendremos que implementar la siguiente funcionalidad:

- Una base de datos de los productos que queremos comercializar en línea.
- Un catálogo en línea de productos, enumerados por categoría.
- Un carro de la compra para almacenar los artículos que el usuario quiere adquirir.
- Una secuencia de comandos de caja que procese los detalles del pago y del envio.
- Una interfaz de administración.

EL_HOLISTICO@HOTMAIL.COM - 2006

El problema

Probablemente recuerde la base de datos.Book-0-Rama que desarrollamos en capítulos anteiiores. En este proyecto, implementaremos la tienda en línea de Book-O-Rama *y* haremos que funcione. Los requisitos de este sistema son los siguientes:

- Necesitamos una forma de conectar la base de datos al navegador del usuario. Los usuarios deben poder examinar los artfculos por categoría.
- Los usuarios deben poder seleccionar artículos del cat61ogo para su posterior adquisición. Tendremos que realizar el seguimiento de los artfculos selecdonados.
- Cuando hayan finalizado sus compns, tendremos que sumar su pedido, procesar los detalles de entrega *y* realizar el pago.
- Tambim tendremos que diseftar una interfaz administrativa para el sitio de Book-0-Rama para que el administrador pueda ai\adir o modificar libros *y* tu categoríu del sitio.

Componentes de la solución

:A continuación analizaremos tu soludones de cada uno de los requisitos expuestos anteriormente.

Generar un catálogo en línea

Ya disponemos de una base de datos para el cat61ogoBook-0-Rama. Sin embargo, para esta aplicación seguramente necesitaremos modificar *y* aftadir algunc:is elementos, como por ejemplo ai\adir categorías de libros, como indicamos en los requisitos. También necesitamos incluir en la base de datos existente derta información sobre envíos, direcciones,detalles de pagos, etc. Ya sabemos c:6mo diseftar un interfaz en una base de datos MySQLpor medio de PHP, por lo que esta parte de la solución será muy sencilla.

Es recomendable utilizar transacciones para completar los pedidos de los clientes. Para ello, tendremos que convertir tu tablu de Book-0-Rama para que utilicen el motor de almacenamiento lnnoDB, un proceso muy sendllo.

Realizar el seguimiento de las compras de un usuario mientras compra

Disponemos de dos formas de realizar el seguimiento de tu compru de un usuario mientras éste compra. Por un lado, podemos ai\adir sus selecdones a la base de datos *y,* por otro, u~ una variable de sesi6n. maso de una variable de sesión para realizar el seguimiento de tu selecdones de una página a otrll resulta m4s sencillo de elCribir ya que no tendremos que consultar mnstantemente la base de datos para obtener dicha informad6n. También evitaremos acumular en la base de datos una gran cantidad de datos inservibles de uiuarios que süiípleni.iñtehan estado navegando *y* después se lo han pensado mejor.

Por ello, necesitaremos diseftar una variable de sesión o un conjunto de variables para almacenar 1uselecciones de un usuario. Cuando éste iennmede reálizar sus compru *y* efectúe el pago, ai\adiremos esta informad6n a la base de datos como registro de la transacción.

También podemos utilizar estos datos para mostrar un nsumen del estado actual del carro en una esquiJla de la f"gina, para que el usuario sepa en cualquier momento cuánto puede gastar.

lmplementar un sistema de pago

En este proyecto, sumaremos el pedido del usuario *y* anotaremos los detalles de entrega. No procesaremos los pagos. Existe una gran cantidad de "91,emu d~ pago disponibles, cada uno con una implementación diferente. Escribiremos una fundón ficticia que se pueda reemplazar con una interfaz en el sistema selecdonado.

Los sistema de pago se suelen venderen áreas geogr6fica mú especfficu que este libro. Por lo general, el fundonamiento de lu distintu interfaces de procesamiento en tiempo real es muy similar. Tendrá que contratar una cuenta con una entidad bancaria en función de las tarjetas que acepte. Su proveedor de sistema de pago especificará los parámetros que debe puar a su sistema.

El sistema de pagos transmitint sus datos al banco *y* devolverá un código correcto o uno de los distintos tipos de códigos de error. A cambio de pasar nuestros datos, el sistema le cargant una cuota fija o anual, uí como una comisión en función del .número o del valor de lu transacciones. Algunos proveedores cobran incluso por transacciones anulado.

El sistema de pagos que seleccione necesitanl informad6n del usuario (por ejemplo, un número de tarjeta de endito), informad6n que noe identifique (para espedficar en qué cuenta se va a realizar el cargo) *y* la suma total de la transacción.

Podemos determinar el total de un pedido a partir de la variable de sesión del carro del usuario. Registraremoslos detalles del pedido final en la base de datos *y,* al mismo tiempo, nos desharemos de la variable de sesión.

Diseñar una interfaz de administración

Además de todo esto, diseftaremos una interfaz de administradón que nos permitint aftadir, eliminar *y* modificar libros *y* categorías en la base de datos.

Una modificadón muy habitual que podemos realizar es cambiar el precio de un producto (por ejemplo, para ofertas especiales o rebaju). Esto significa que al alma-

EL_HOLISTICO@HOTMAIL.COM - 2006

cenar el pedido de un cliente, también tendremos que almacenar el precio que haya pagado por un artículo. Nuestra contabilidad seria desastrosa si los únicos registros que tuviéramos fueran los artículos pedidos por cada cliente y el precio actual de cada uno de ellos. Esto también significa que si el cliente devuelve o cambia un artículo, se le abone la cantidad correspondiente.

En este ejemplo no diseñaremos una interfaz de seguimiento de todos los pedidos. Puede añadir una que se ajuste a sus necesidades a este sistema básico.

Presentación de la solución

A continuación combinaremos todas las piezas. El sistema cuenta con dos vistas básicas: la vista del usuario y la del administrador. Después de determinar la funcionalidad necesaria, definimos dos diseños de flujo del sistema, uno para cada vista, que mostramos en las figuras 27.1 y 27.2 respectivamente.

En la figura 27.1 se muestran los enlaces principales entre las secuencias de co~d~ de la parte del usuario del sitio. En primer lugar, el cliente accede a la página principal, en la que se enumeran todas las categorías de libros del sitio. Desde aquí, puede acceder a una categoría determinada y, tras ello, a los detalles conaetos del libro.

Figura 27.1. La vlstll de usuario del sistema Book-0-Rama permite a los usuarios examinar los libros por categorfa, ver detalles de los libros, alladlrlos al carro y comp,artos.

A cada usuario se le proporciona un enlace para añadir un libro concreto a su carro. Desde el carro, podrá salir de la tienda en línea.

En la figura 27.2 puede ver la interfaz de administración. Tiene más secuencias de comandos, pero no gran cantidad de código nuevo. Estas secμendas de comandos permiten al administrador registrar y añadir categorías y hbros.

Figura 27.2. La vista de administnlclón del sistema Book-0-Rama petmite alladlr, modificar y eliminar libros y categorfas.

La forma más sencilla de implementar la modificación y eliminación de libros y categorías consiste en mostrar al administrador una venión ligeramente distinta de la interfaz de usuario en el sitio. El administrador puede seguir naveg~do por libros y categorías pero en lugar de tener acceso al carro de la compra, podrá acceder a un determinado libro o categoría y modificar o eliminar dicho libro o categoría; Si conseguimos que ambas secuencias de-comandos funcionen tanto con usuarios como con administradores, nos ahorremos tiempo y esfuerzo.

Los tres módulos de código principales para esta aplicación son los siguientes:

- Catálogo
- Carro de la compra y procesamiento de pedidos (los hemos agrupado porque están íntimamente relacionados)
- Administración

Como en el proyecto anterior, tendremos que generar y utilizar una serie de bibliotecas de funciones. En este proyecto, utilizaremos un APl de funciones similar al anterior. Intentaremos agrupar las partes del código que generen HTML en una misma biblioteca para cumplir el principio de separación entre lógica y contenido y, lo más importante, para que nuestro código resulte más fácil de leer y de mantener. También tendremos que realizar algunos cambios en la base de datos Book-0-Rama. Hemos cambiado el nombre de la base de datos por book_sc: para distinguirla de la que creamos en capítulos anteriores.

Todo el código de este proyecto se incluye en el CD-ROM. En la tabla 27.1 encontrará un resumen de los archivos de la aplicación.

EL_HOLISTICO@HOTMAIL.COM - 2006

T•bl• 21.1. Archivos del• •pllC8Clón carro de la compra.

index.php	C8lillago	NglnIl pmc:¡p.IdlllllIlo pera loe IIIUIfIol. Mueslrll une Hite de In cetegorla del lllteme.
•h-_cat.php	C8lillago	Mueatre el lllueno toda.m llbrolde une clelannlnade cetegcxte.
•~_book.php	C8lillago	Muatrll el 111uerto loe CNtellell de un llbrv 811 CDIICnllo
•h-_cart.php	C-delacompre	Muestre el uaueno loe co111e111dos de IU cano. Tembl6n .. UlIIID pera elledlrerll-culol el cem,.
checkout.php	Cerro de lacomp,e	PreMnteelUIUSrtoloedeleleadllpedldo. ObUent deloedeenvlo.
purchaH.php	Cerro de la compre	Oblllllned ..... depegodll Uluerto.
procaee.php	Cerroele .. compre	P,_.. loe d..... ele pego y elledll el pedido e le.,... ele delos.
logia.php	Admlnlalrec:16n	P-lle el edmlnillnldcrconec:t.rM pare rullzercembloe.
logout.php	Admlnlancl6n	o.-i.ctll el llllllarloedmlmlredor.
admia.php	Adnd111alrecl6n	Men6 prlnclplll ele edmlnilllrecl6n.
challge_paeaword_fora.php	Admlnlalrec:161'l	Fannularto pera que el edmlnlltradorpue-de cemblareuconir.elle.
challge_paaaword.php	Admlnillrecl6n	Cemblelll conlreMlledel eclmlnlalredor.
iaeert_category_fora.php	Admlulalrec:16n	Fannulerto pera que el ecl,•lletlecloi pue-dHlledlr-nuevecellgclrleelei.ede deloa.
inaert_category.php	Admlnlllrecl6n	Alllde ... la nueve celegcrie e la bllM de delOI.
iaeert_book_form.php	Admlulalrec:161'l	Fannulertop .. queeledmlnlllredorpue-dll elledirun ..._ llbroellllteme.
inHrt_book.php	Admlullnc:16n	Nlecleel-llbn> ela bueeledlllos.
edit_category_form.php	Mrmlllllrec:l6,i	Fonnularlo perequeelednllnllndor pue-de modlflcer une cetegorf&
edit_category.php	Admlnlllrec:l6n	Acluellzll •• eólgorl• en le bllM de dlllOI.
edit_book_fora.php	Admlnillraci6n	Fannularto pare que el edmlldllradormo-dlllque loe deie... ele un llbro.
edit_book.php	Ad,nltlllrec:l6,i	Aclulllzll el Nllro en la i.e eledllloa.
delete_category.php	Admlnillnlcl6n	Etlmkle unecetegcxtedelebeleeleclelOI.

dalete_book.php	Adl1,Mec:l6l1	Elmlne un llln, de le b.. de cleloa.
book_ac_fna.php	Func:lonN	Colecc:l6nde ...._delnc:luli6nde ....
adllill_faa.php	func:ionN	Coleccl6ndefunclol'IN uliladMpor... decomendoaecl11,11111e11.-.
book__fu.php	Func:lonN	Coleccl6nclefunc:lone•pareelmecel•y NCUpererdelOIIClbre llbn,e.
order_faa.php	Func:ionN	Coleccl6ndefunclone•pere..._•., rea,pererd.iaasobre pedldoe.
output_faa.php	Func:lonN	Coleccl6ndefunclonfl pera..... HTML
data_valid_fu.php	Func:lonN	Colecc:l6ndefuncloneaparevellder.,... dNdedelOI.
clb_fu.php	Func:lonN	Coleccl6n de runctone• pare conec:t.rM e le blIN de deliOI book_ac.
uaer_auth_faa.php	Functon.	Coleccl6n ele funclo.... ...,..._
book_ac.9ql	SQL	SCI. pera dellnlrla baMdedelOlbaok_ec.
populate.aql	SQL	SQL .,.. .nedlr deliOI de ajllmplo e le bale de delOlbook_ac.

A continuación veremos la implementación de cada uno de estos módulos.

160 4. Manipular cadenas y expresiones regulares

Esta aplicaci6n llene una gran cantidad de código. Gran parte del mismo Implementa funcionalidad que ya hemos visto (en especial en el capitulo anterior) como por ejemplo el almacenamiento y recuperación de datos de la base de datos, y la autenticación del usuario administrativo. Nos detendremos brevemente en este código, ya que nos centraremos en las funciones del cano de la compra.

Para que el código de este proyecto funcione como está escrito, tendré que activar las comillas mlglcas. SI no lo ha hecho antes, tendré que aftadir addslashes (› a los datos destinados a la base de datos MySQL. Lo hemos utilizado como método abreviado 6til.

Puede activar ias comillas mlglcas en cada directorio de un archivo •htaccess por medio de la directiva

php_value .. gic_quotea_gpc on.

EL_HOLISTICO@HOTMAIL.COM - 2006

Implementarla base de datos

Como hemos mencionado anteriormente, hemos realizado algunos cambios en la base de datos Book-0-Rama que presentamos en un capitulomterior. El código de SQL para crear la base de datos book_sc se muestra en el listado 27.1.

Llmdo 27.1. boOk_sc.sql. Codigo SQL para crear la base de ct.tm book_sc.

```
create databa•• boOk_•c;

ll•• book_•c;

create table C118t011er•
(
   cu•t011erid int unaigned aot null auto_incr...nt pri..ry key,
   na- cbarl60! Pot null,
   addr••• c:har(IOI not null,
   city char(JOI not null,
   •tate char(20I,
   &:ql cbar(lOI,
   country char(201 not null
   tY,!i19dlllIODB

cr~at• table order•
(
   ~..J.AC.      •m•ign•d nar. Dlll a11to_incXW11t ai,'.ian...QL
   cwato.arid int unaigned not null,
   atirount float(6,21,
   date-date not null,
   order •tatu• c11&r(10!,
   •hip_;... charl601 not null,
   •bip_addre•• char(IOI not Dllll,
   ahip city char(JOI not null,
   •hip=•tate cbar(20),
   •hip_aip char(lOI,
   •hip_country char(201 not null
   type•InnoD8

create table book•
(
   iiil:m char(lll not null pri..ry key,
   ~:e char(lOOI,
;~tl-:_,¡,c:;har(lOO),
   ~d.al ~t —igned,
   prioe'"float(4,2) not null,
   deiicription varc:har (255)
) tnie-I_.,.

e~ table categori••
('~'!'
   catid int unaigned not null auto_incr...nt prima:y key,
   catna.. char(60) not null
   type•InnoDS
```



```
create table ordar_itema
(
   orderid int —igned not null ref•r•nc•• ordar•lorderidl,
   iabn char(lll not null referance• booka(iabnl,
   it.._price float(4,21 not null,
   ciu-zatity tinyint un•ivned not Dllll,
   pri..ry key (orderid, iabnl
   type•IDDODB

create tableadain
(
   ll••rna.. char(161 not null pri..ry key,
   P&••-rd cbar(40) not null
)i

grant ••lect, iuart, update, dalete
on book •c.•
to book:•celocalbo•t identified by •pa•-rd'l
```

Aunque la interfaz Book-0-Rama original era totalmente v6lida, ahora disponemos de algunos requisitos adidcmales que tendnn que estar disponibles en linea. A continuadón se enumeran los cambios que hemos realizado en la base de datos original:

- Hemos aftadido m'5 campos de di.recci6npara los clientes. Ahora es m'5 importante ya que queremos crear una aplicadón m'5 realista.
- Hemos añadido una clirecáón de envío al pedido. Puede que la direcdón de contacto de un cliente no sea la miliftl" que i.-dtrecdiSn "Hft\'f<>, sobre todo si está utilizando el sitio para comprar un regalo.
- Hemos añadido una tabla de categorías y un Id. de categoriua la.tabla de libros- Al clasificar los libros en categorías resultan m'5 sencillo navegar por el sitio.
- Hemos añadido item_price a la tabla order_itema para indicar que el precio de un artículo puede cambiar. Queremos saber lo que costaba cuando lo adquirió el cliente.
- Hemos añadido una tabla admin para almacenar los detalles de conexión y la contrasei\a del administrador.
- Hemos eliminado la tabla rev:l.ewa (p~ede aftadirla como extensión al proyecto). En su lugar, cada libro dispone de un campo de descripción en el que se incluye una breve informadón sobre el mismo.
- HemOI(aDlbiado el motor de almacenamiento a InnoDB,para poder utilizar claves secundarias y transacciones al introdudr información sobre los pedidos de los clientes.

Para configurar esta base de datos en nuestro sistema, es necesario ejecutar la secuencia de commdos book_sc. aql a través de MySQLcomo usuario raíz:

```
,ayaql -u root -p < book_•c.aql
```

(Tendr6 que indicar su contrasei\a raíi)

EL_HOLISTICO@HOTMAIL.COM - 2006

Antes de nada, tendrá que cambiar la contrasei\a del usuario book_ac: por otra distinta. Si cambia la contrasei'la en book_sc: también tendrá que cambiarla en db_fnl!_:php (como veremos en breve).

Hemos incluido tánibié it"un archivo ae datos de ejemplo con el nombre populate. sql. Puede aftadir los datos a la base de datos si lo ejecuta desde MySQL de la misma forma.

Implementar el catálogo en línea

Esta aplicación cuenta con tres secuencias de comandos de catálogo: la página principal, la página de categorías y la página de detalles de libros.

La página principal del sitio se genera por medio de la secuencia de comandos index. php, cuyo resultado se muestra en la figura 27.3.

Figura 27.3. La pégina Inicial del sitio enumera las cat9gorfaa de libros en venta.

Comprobará que, además de la lista de categorías del sitio, hay un enlace al carro de la compra en la esquina superior derecha asf como información resumida d_e los contenidos.del mismo. Esto aparecerá en todas las páginas mientras el usuano na,·ega r real~~ sus compras. Si el usuario pulsa sobre una de las categorías, accederá a fa pagana de categorías, generada por la secuencia de comandos ahow_c:at •php. La página de categorías de la sección de libros sobre Internet se muestra en la figura 27.-l..

Figura 27.4. Cada uno de 101 libro, de una c:alllgorfa Incluye IIIIII rotografla.

Todos los libros de la categoría Internet se enumeran en forma de enlace. Si el usuario pulsa sobre uno de estos enlaces, accederá a la página de detalles del libro. En la figura 27.5 puede ver una página de detalles de un libro.

En esta página, asf como el enlace Vlew Cart, vemos un enlace Add to Cart que permite al usuario seleccionar un articulo. Lo describiremos m6s adelante cuando veamos cómo se genera el carro de la compra.

A continuación analizaremos detalladamente cada una de estas secuencias de comañdos.

Enumerar categorías

La primera secuencia de comandos, index. php, enumera todas las categorías de la base de datos. Se muestra en el listado 27.2.

Uataclo 27.2. Index.php. Secuenc:la de c:omandoa para generar la péglna inlc:lal del 1ltlo.

```
c?php
   require ('.booJc_ec__fu.pbp'l1
   // Bl cerro de le c0lll)re neceeita  •••ionee;  inicie una
   ••••ion etart O I
   do_htal)ieaderl'welcOM   to Book-0-Rau'  l I

   echo 'cp>Pleaae  chooee a category:c/p>';
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
// obteiaga categorlH de la ban de dat-
$cat_array • get_categorieall;

// 1111 6atrela• como enlaces a laa P'ginaa del catllogo
diaplay_categoriea(Scat_array);

// ai ae conecta cciio adlliniatrador. muestre enlaces para alladir,
// eliainar y modificar categorla•
ifliaaetlS_SESSIOJl('adaia_uaer•J)I
{
   diaplay_buttoa('adllia.pbp•, 'adain-meau•, 'Adaia Meau'I;
}
do_htal_footerll;
```

Figura 27.5. C.. libro cuenta con una página de cleWIes en la que • muNlnl información adlc:ional asf como una extensa descripción.

En primer lugar, l• secuencia añade book_sc. php, el archivo que incluye todas la bibliotecas de funciones de esta aplicación.

Tras ello, es necesario iniciar una sesión, para que la funcionalidad del carro de la compra sea efectiva. Todas las p4gu,as del sitio utilizarút esta sesión.

Hay varias llamadas • funciones de representación H1ML como do_html_header() ydo_btml_footer() (ambasinc:luiduenoutput_fna.php). Tambim tenemos un código que comprueba si el usuario se ha conecta.do como administrador y qu~ le ofrece distintas opciones si lo ha hecho, como veremos en el aputado de funciones de administración. La pute má importante de esta secuencia de comandos es:


```
// obtenga categorlaa de la base de dat-
$cat_array • get_categoriea(I;

// mu6atrelaa c:oao enlace• a laa plginaa del catllogo
diaplay_categoriea!$cat_array!1
```

Las funciones get_categories O y diaplay_categories () se encuentran en las bibliotecas de funciones book_fn• .php y output_fns.php, respectivamente. La función get_categories {) devuelve una matriz de las categorfas del sistema que, tras ello, se pasa a display _categoriea (). En el listado 27.3 podemos ver el código de get_categoriea ().

Llst.clo 27.3. Función get_c:megories() de book_fns.php. Ella funcl6n recupeni una liltll de categorflls de la bae de datos.

```
func:tion get_c:categorieall
{
   // c:oaaulte la base de datoe para obtener una lista de cetegorla•
   Scoaa • db_coanect(I;
   Squery • •aelect catid, cataa..
         froa categoriea•;
   Sreault • ec:oan->queryl$queryl;
   U 11$naultl
      retura false,
   saua cata• esreault->nua rova,
   if (Snu• cata• •01 -
      returñ false:
   Sreault • db_reault_to_array(Srell:ultll
   retura Sreault;
```

Como puede comprobar, esta fundón se conecta a la base de datos y recupera una lista de todos los Id. y nombres de ategorfll. Hemos escrito y utilizado una funciónde nominadadb_re•ult_to_array()queseencuentraendb_fn•.php. Esta función se muestra en el listado 27.4.

Adopta un identificador de resultados MySQLy devuelve una matriz de errores indexada numéricamente en la que cada fila es una matriz asociativa.

Ust.clo 27.4. Funci6n db_result_to_a~) de db_fns.php. Func:16n que c:onvterte un IdentlflcadOr de resultados MySQL en unai matriz de resultados.

```
¡uaction db_reault_to_array!Sreaultl

   Srea_array • array!!l

   for 1$couat•01 Srov • $reault->fetcb_aaaoc:(I; $count++I
      Sraa_array[Scouatl • Srow;

   retura Srea_array;
```

En nuestro caso, deYolveremos esta matriz de nuevo• index.php donde la puaremos a la funéiónd isplay _categoriH () de output_fna. php. Esta fun-

EL_HOLISTICO@HOTMAIL.COM - 2006

d6n muestra cada categorfa en forma de enlace a una p6gina que contiene loe libros de diclia categorfa. En el listado 27.S podemos ver el código de la función.

Listado 27.S. Func:16n dlsplay_c8111gorlea() de output_fns.php. Func16n que ffIUltIlra una matriz da cetegorlaa como llsfll de enlacN a dichas catagorfaa.

```
funetion dil1¡1lay_categori.. ($cat_array)
{
   if llia_array($cat_array)1
   {
      echo 'llo categori•• currantly aYailablacbr />'1
      return1

   echo '<ul>'1
   foreach 1$cat_array aa $row)
   {
      $url • •abow_cat.pbp?catid•'.($row(•catidJ11
      $titla • Srow(•catnama'J;
      echo '<li>'1
      do_ht•l_urll$url, $titla11
      echo '</li>

   echo •c/ul>'1
   echo •chr />',
```

Esta función convierte cada una de las categorfas de la base de datos en enlaces. Todos los enlaces llevan a la siguiente secuencia de comandos, ahow_cat .php, J,ero cada uno tiene un par6metro distinto, el Id. de la éategorfa o cat id (se trata de un número exclusivo generado por MySQL y que se utiliza para identifi~arla categorfa).

Este parimebo a la siguiente secuencia de comandos determinarA en qu6 categoría nos encontramos.

Enumerar los libros de una categoría

El proceso para enumerar libros de una categorfa es muy similar. La secuencia decomandosencargadadeelloes show_cat .php,que podemos ver en el listado27.6.

Uatado 27.1. llhow_cat.php. Esta aeaaenclll de comandos muestra loa libros de una detltrmlnada catapfa.

```
c?php
   requira 1 'book_ac_fna.pbp' 1;
   11 El carro da la compra nacaaita •••ioaaa; inicia una
   ••••ion"atart 111

   $catid • S GIIT('catid'11
   $na.a• gat_category_nama1$catid11

   do_html_haadar1$n-11'
```



```
   // obtenga la informaciOn dal libro da la ba•• da datoa
   $11ook_array • gat_bookal$~atidl;
```

```
   // al •• conecta co.a edlliniatrador, •••tJa .•.!_lacea para aladir
   1/ y ali•inar libro•
   }f(iaaat1$_SESSI01f('edllin_uaar•J)I

      diaplay_button1'indax.php', 'continua', 'c:ontinua Shopping' 1,
      diaplay_button( •adlu.n.pllp', •.-1n--u•, 'Adllin Nenu' 1,
      dil1¡1lay_button1•adit_category_form.pllp?catid•$catid•,
         •adit-category•, •lrdit catagory'11

   ala•
      diaplay_button1'indax.pllp', •continua•allopping', 'Continua Shopping•J,

   do_html_footero,
?>
```

Esta secuencia de comandos tiene ~ estructura muy similar a la de la p6gina de Indice, a excepción de que recupera libias en lugar de categorías.

Como de costumbre empezamos con aeaaion_atart () y, tras ello, convertimos el Id. de categorfa que hemos obtenido en IDI nombre de categorfa por dlldio de la funcidn get_category _na• (), como se indica a continuadón:

```
$nama • gat_category_11&11e1$catidl1
```

Esta funci6n busca el nombre de la categorfa en la base de datos. Su código se muestra en el listado *27.7.*

U.lado 27.7. Funcl6n get_catagary_name() de book_fns.php. Esta funcl6n convlerbt un Id. de cal9gorla en un nambnt de calegorfa.

```
func:tion gat_category_na..cscatidl
{
   JI consulta la bHa da dato• para obt-r al IICIIIbn da un Id. da catagocda
   $catid • intvall$catidl;
   $coan • dh_connectll1
   Squary • 'aalact cat-
            from categoriaa
            wllara catid • $catid"1
   $raault • esconn->queryl$quaryl,
   if 11$reaultl
      return falaa;
   $$DU11_cata • e$reault·>nU11_rowa,
   if 1$nU11_cata • •OI
      return falaa;
   $row • $reault·>fatch_objactll1
   raturn $ ~ ·>catna..,
```

Una vez recuperado el nombre de la categoría, podemos representar un encabezado HTML y proceder a recuperary enumerar los libros de la base de datos que se corresponden con la categorfa seleccionada, como se muestra a continuación:

EL_HOLISTICO@HOTMAIL.COM - 2006

```
$book_array • gec_bookeC$cacidl1
diaplay_bookal$book_arrayl,
```

Lasfuncionesget_book•() y diaplay_booka O sonmuysimilaresalasfunciones get_categoriea () y diaplay _categories (), por lo que no las describiremos. La única diferencia es que estamos recuperando información de la tabla de libros en lugar de la de categorías.

La función display _book• () proporciona un enlace a todos los libros de una categoría por medio de la secuencia de comandos show_book.php. De nuevo, a cada enlace se le ai\ade un parámetro como sufijo. En esta ocasión es el ISBN del libro en cuestión.

En la parte inferior de la secuencia de comandos show_cat .php, verá que hay un código para representar funciones adicionales si se conecta un administrador, como veremos en el apartado sobre funciones administrativas.

Mostrar detalles sobre un libro

La secuenda de ~mandos ahow_book. php adopta como parámetro un ISBNy recupera y muestra los detalles del libro en cuestión. El código de esta secuencia de comandos se incluye en el listado 27.8.

Listado 27.8. show_book.php. Esta sea,ancia de comandosmunlra los detalles de un determinadolibro.

```
c?p!!,
   require l'book_•c_fna.php'I;
   II Bl carro de la c011pra neceaita aeaione•, inicie una
   •e•aion_~tartll,

   $i•bn • $_GBT['i•bn•J;

   // obtenga este libro de la baae de datos
   $book • get_book_decailsl$iabnl;
   do html header($book['title')I;
   diaplay:book_deta11•($bookl;

   // defina un URL para el •bot6n de continuac16n•
   $target • 'index.php';
   1f($book[•catid')I
   (
      $target • '•how_cat.php?catid-'.$book(•catid');
   J
   // ai - conecta e- adaiaiatrador, —e•tre ealac•• para modificar libro•
   if( cbec:k_adllia _uaerl! !
   (
      cli•play_buttoa(•edit_book_fora.pbp?isba-$.ulla",'edit-it-·, 'Bclit n-· 1;
      display buc.coa(•admia.php', •amia-—au•, 'Adaia lleaa'I;
      clieplay:buttoa($target, •c:ontinue•, 'ContiJaue•);
   J
   els•
   (
      di•play_buttoa(•show_cart.php?new•$iaba", 'add-to-cart', 'lldd '
            .$book('title'J.• To Ny Shopping cart'I;
      di•play_button($target, •continue-sboppiJag•, •contiaae Slulpping•);

   do_html_footerCI,
```

Esta secuencia de comandos realiza el mismo tipo de acciones que hemos descrito en p6ginas anteriores. En primer lugar iniciamos la sesión y, tras ello, utilizamos

```
$book • get_book_detaila($i•ba);
```

para obtener de la base de datos la información del libro y

```
di•play_book_details($book);
```

para representar los datos en HTML Conviene ~clonar que diaplay _book_ details () busca un archivo de imagen para el libro como imagea/$isbn. jpg. Si este archivo no existe, no se muestra ninguna imagen. El resto de la secuencia de comandos configura la navegación. Un usuario normal tendrá las opdones Contiriue Shopping, lo que le devolverá a la página de categorías y Add to Cart. que añadirá el libro al carro de la compra. Si el usuario se ha conectado como administrador, dispondrá de opciones düerentes, que analizaremos mis adelante. Con esto se completan los elementos básicos del sistema de cat61ogo.A continuación describiremos el código correspondiente a la funcionalidad del carro de la compra.

Implementar el carro de la compra

La funcionalidad del carro de la compra se centra en la variable de sesión cart. Se trata de una matriz asociativa que utiliza ISBNcomo claves y cantidades como valores. Por ejemplo, si aftadimos una sola copia de este libro al carro de la compra, la matriz contendrá

```
0672317842-> 1
```

Es decir, una copia del libro con el ISBN0672317842. Cuando ai\adimos artículos al carro, se añaden a esta matriz. Cuando revisamos el carro, utilizamos la matriz cart para buscar los detalles completos de los artículos en la base de datos.

También utilizamos otras dos variables de sesión para controlar la pantalla del encabezado que muestra Total ltems y Tolal Price. Estas variables son itema y total _price respectivamente.

Utilizar la secuencia de comandos show_cart.php

En primer lugar !'eremos cómo se implementa el código del curo de la compn por medio de la secuencia de comandos show_cart •pbp. Se trata de la secuencia

EL_HOLISTICO@HOTMAIL.COM - 2006

de comandos que muestra la p4gina a la que accedemos si pwsamos sobre'CWll-quier enlace *View* cart o Add to Cart. Si invocamos ahow_cart . php sin parúnetros, podremos ver los contenidos de la misma. Si la invocamos con un ISBN como parúnetro, se aAaclid al carro el artfculó qüétenga dicho IS BN.

Para comprenderlo, ffjese en la figura 27.6.

Figura 27.1. La secuencia de comandos 1how_cart.php IIn par6melrol muestra CInlcamente el contenido de nuestro cano.

En este caso, hemos pulsado sobre el enlace VI~ Cart con el carro vado, es decir, no hemos seleccionado ningún artículo para su compra.

En la figura 21.7 podemos ver el carro después de seleccionar dos libros. En este caso, hemos accedido a esta p4gina tras pulsar sobre el enlace Add to cart de la página ahow_book. php del libro *PHP and* MySQl. *Wtb lñwlopnrent*. Si se fija en la barra de URL, verá que en esta ocuión hemos invocado la secuencia de comandos con un parámetro. El parámetro es new *y* tiene el valor 0672317842, es decir, el ISBNdel libro que acabamos de aftadir al carro.

Comprobar6 que desde esta página tenemos dos opciones. Hay un botón Se.ve Cwages que podemos utilizar para cambiar la cantidad de artículos del carro. Para ello, podemos modificar las cantidades directamente *y* pulsar Se.ve Omges. De hecho se trata de un botón de envio que nos lleva de nuevo a la secuencia de comandos ahow_cart . php para actualizar el carro.

Ademú, hay un botón Go to Cbec:kout que el usuario puede pulsar cuando quiera salir, como veremos más adelante.

Figura 27.7. La secuencia Ihow_cart.php con el nuevo parémetro aÁade un nuevo 91tlculo al carro.

Por el momento, nos detendremos en el código de la sec:uenda de comandos ahow_cart . php, que se muestra en el listado 27.9.

Listado 27.t. Ihow_c:art.php. Esta secuencia de comandos controla el carro de la compra.

```
c?php
   require (•book_ec_fnl1.php');
   // Bl carro de la CC11pra neceait.- aeaionea; inicie una
   ••••ion_atart O /

   • Snew • s_on[•newJ,

   if($newl
   {
      //nuevo el...nto aeleccionado
      if(liaaet($ _SBSSIOll['cart•))I
      {
         $ SSSSION[•cart') • arrayll:
         $-SUSIOll['iteM') • 0;
         s:sssSIOll['total_price•J •'0.00';
      }
      if(iaaet1$ SUSION(•cart•J ($naw)II
         $ SUSION[•cart•J l$new)++;
      elae-
         s_sSSSIOll['cart•) ISnewJ • 1,
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
$_8...ICIIJl'tocal_price'J   • calculata_priceC$_SBSSIOll(,cart'JJ,
s_ssssIOII ('ita ..•J • calculate_1t...C$_SSSSI<m('cart,Jl:


ifliaaetl$_POSTC... ve,lll
(
roreach <t_... SICIIJC'cart'J aa $iabn ... Sqtyl
(
ifl$ POSTC$iabaJ• •'O'I
ui.etlS_SBSSIOll(•cart•J ($iabnJ1l
eln
s_sSSSJOIIC'cart•J CSiabal • S_POSTCSiabDI:
|
s_sBSSICIIJ('total_pric•'J • calculate_price|$_5..,IOIIC'cart'11 s
s_ssssIOll('it ...•J • calculate_it...cs_ssss1o•c'cart'lll


do_htal_lwacierl'Tour eboppi119cart'l;

iflS_SBSSIOll('cart'J• ..rray_count_valueeCS_SSSSI<m(•cart'lll
diaplay_cartl$_SSSSIOll(•cart•JJ,
•]••
(
ecllo •cp>Tbere are ao it... in yoar cartc/p>':
•cho 'ciar /•';
}
Stair:vet • •i.Jldex.php•;

!! ai acaba 4• ..dadi.t....lll
// caugoda
ifl$uwl
(
Sdetaila • vet_bOo..detailars-wl
iflSdetaila(•catid'll
$ta1:9et • 'abow_cat.php?catid•'.$detaila(•catid'l1
}
dieplay_buttoa(Staz,¡et, •coatinue-aboppiag', 'continua Sbopping•);

/l utilice eato ai ba coll!igurado SSL
// SpatJa • ,_sava('PRP_SSLP'I;
11 $aerver • s_sSRVBJl('SSJIVEII_IIAIIS'l1
/I $path. etr~replacel'ahow_cart.php', , $pathl
11 diaplay_buttoa('httpa://'.$aerver.$path.'ch•cfi.out.php',
'go-to-checfi.out', 'Go To Cllecfi.out'l

// ai ao ha coaUgurado SSL, utilice el aiguient• c6digo
dieplay_butt0IIC •cbaefi.out.php', •go-to-checfi.out', 'Go To Cllecfi.out' I:


do_htal_footerCl,
```

En esta secuencia de comandos hay tres partea principales: la representación del carro, la inclusión de los ardculos en el carro y la aplicad6n de los cambios. Deecribiremos cada una de estas partes en los siguientes apartados.

Ver el carro

Independientemente de la ptgina de la que vengamos, podremos mostrar el contenido del carro. En el cuo básico, cuando un usuario acaba de pulsar Vlew Cut. la única parte del código que se ejecutam es la que mostramos a continuación:

```
if 1$_S..,I011['cart•J ..array_count_valueel$_SSSSIC9('cart' J11
diaplay_cartl$_SSSSIOll('cart'lll
•]••
(
echo 'cp..There are DO 11:... in yaur cartc/p..':
ecbo 'ch% j,,',
```

Como puede comprobar, si tenemos w, cuadro con algdn tipo de contenido, invocaremos la funci6n display_cart (>. Si el cano esd vado, mostraftlllos el correspondiente mensaje al usuario.

La fund6n display_ cart () simplemente imprime los contenidos del carro en un formato HTML legible, como puede comprobar en las figuras 27.6 y 27.7. El c6digo de esta función se incluye en output_fns. php que m~os en el listado 27.10. Aunque es una función de representac:i6n.es bastante compleja, razón por la que la induimos a continuación.

Listado 27.10. Funcl6n clilplay_cart()ele OUlpUt_fns.php. Eú func:i6n eplk:a un formlllo a los contenldoa del cano ele la comsir- y los imprime.

```
f1111etioa diaplay_cartl$cart, $changa • true, $1-ve• • 11
(
// .... acre 109 artlculoa del carro de la-compra
11 opcioaal..nte permita callbioa (true o falael
11 opcioaal..nte i~luya illAgen•• (1 - •l, 0 - DO)

echo •ctlll:tl• border•"O" widl:h-o1001" cell•paciag•"O""
cform aetiOD•"•bow_cart.php• ..tlloda"poat•..
ctr..cth colmpaa•"' (1+$1.....•l .•• bgcolor•"lccccccc•..Iteac/tla..
cth bgcolor••tcccccc•.,pricec/th ..cth bgcolor•"lccccccc•>,Quantityc/th ..
cth bgeolor••tcccccc•-Totalc/th ..c/tr..';

//eueetre cada al...nto COIIO fila de waa tabla
foreach CScart a• $1abn .,, $qtyl
(
$book • get_book_detail•l$iabDll
echo •ctr> •1
1r1s1...,.•• •t:naal
(
ecbo 'ctd ali911-•le~t•..'l
if lfile_exi•t•l"iaag••l•i•bD.jpg•ll
(
$aisa • GetI..,-Sise('iaagee/' '$ia!ln.''jpg'J,
ifC$aise(OJ..0 .. $aise(ll..Ol
(
echo 'ciag arc••iaage•/• .$1alln.'.jpg• border-0 ',
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
            echo •wiclth •    $eise(0)/1    'height  •  '  ,Seise(1)/3   '  ~·1
      }
      el•e
         echo  'lo1lb•p1·1
      echo ·c/td>';

   echo  •ctd aii9n••left">'1
   echo ·ca  href•"•how_book.php?i•bn•',$i•bn.••,.•.$book('title•J.
      'e/a>  by '.$book(•author');
   echo  'c/td>ctd align••center">$'.nu!IIH!r_format($book(•price•J,     2!1.
   echo  'c/td>ctd align.•center">';
   11 •i penalti—• callbioe, la• cantidadea ae aue•tran en cuadroa de texto
   if  ($chllnge  • • true)
      echo 'cinput type••text•  na..•"Siabn•  value••$qty•  ai••··•J•,.•,
   el••
      echo  $qty;
   echo  'c/td>ctd  align··center•>$'.nllllber_format($book('price•J•Sqty,21
      ·c/td>c/tr>\n';
}
// aueatre el total de fila•
echo  •ctr>
      cth colep&n•"'·  (2+$i ..ge•I  .••  bgcolor••lcccccc•>t.nbap,c/td>
      cth align  •  center  bgcolor•"lcccccc">
         ',S_SISSIOll('itema•J,•
      c/th>
      cth align  •  center bgcolor•"lcccccc">
         $' '.nllllber_for..t($11TTP_SISSIOR_VARS(•total_price•J,   21,
      'c/th>
   c/tr>';
// mueatre el bot6n para guardar callbioe
1f($change   • • true)
{
   echo  •ctr>
         ctd colepan•"'·  (2+$1mage•I   .">t.nbep¡c/td>
         ctd align••center">
            cinput type••hidden•  name••aave•  value••true•>
            cinput type••i ..ge•  erc••i ..9e•/•ave.;han9ea.9if•
               border••O•  alt•"Save  Changea•,-
         c/td>
         ctd>t.nbep;c/td>
      c/tr>';
}
echo  'c/fOrl1>c/table>';
```

El flujo búico de esta función se describe a continuación:

1. Realiza un bucle por cada uno de loa artículos del carro y pasa el ISBN de cada uno a get_book_details (1 para que podamos resumir los detalles de cada libro.
2. Proporcionauna imagen para cada libro, si existe alguna. En este caso utilizamos lu etiquetas HTML de altura y anchura de imagen para reducir el tama1'o de la misma. Esto significa que las imágenes aparecer6n ligeramente

distonionadaspero Jo~te reducidas para poderulflizar)u (siempre puede cambiar el tamailo con ayuda de la biblioteca gd que describimos en un capftulo anterior o generar manualmente im4genes de distinto tamaflo para cada producto).

3. Cada entrada del carro dispcme de un enlace al correspondiente libro. ei- decir, a show_book. php con el ISBNcomo pañmetro.
4. Si invocamos la función con el panmebO change configurado como true (sin configurar adopta true de forma predeterminada), muestra los cuadlOs con las cantidades en forma de formulario con el botón S.ve Oumps en la parte final (al volver a ulilizar esta función despuá de salir, el usuario no debe poder cambiar su pedido).

Realmente esta funci6n no es excesivamente complicada, pero requiere una gran cantidad de trabajo, por lo que deberfa repasarla con atención.

Añadir artículos al carro

Si un usuario ha accedido a la p6gina show_cart. php tras puJ&llbotónAdd lo Cut. tendremos que realizar algunas operaciones antes de pod { fiifostrarlelos contenidos de su carro. En concreto, ser6 necesario aftadir el correspondienteartf· culo al carro, como indicamos a continuación.

En primer lugar, si el usuario todavía no ha atladido artfculos al carro, no tendd un carro, por lo que será necesario aear uno:

```
if(lia8et($11TTP _SBSSIOII_VAJIS(•cart'lll
{
   $_SBSSION['cart'J  •  array!);
   $_SBSSICll('it ...•J  •  0l
   s:sssSIOlll'total_price•J  .'0.00',
```

Inicialmente, el carro esd vado. Una vez audo el carro, podemos ailsdir artf· culos al mismo:

```
ifli•aet(S_SSSSICll['cart•J  (Snew)ll
   s_sssSION('cart·J[SnewJ++;
•l•e
   S_SSSSICll['cart·J(Snewl  •  11
```

En este caso, comprobamos si el artículo ya se encuentra en el carro. Si ya lo esd, incrementamos en uno la cantidad de dicho artfculo en el carro. En caso contrario, atladimos el nuevo artfculo al carro.

Tras ello, Sed necesario calcular el precio y el número total de los artfculos del carro. Para ello utilizaremos Ju fundones calculate_price (1 y calculate_ ite819 (1 de esta forma:

```
s_ssssION('total_price·J   •  calculate_price(S_SSSSIOR('cart•Jll
$_8SSSICR('it ...•J  •  calculate_it...[$_SBSSIOll('cart')I;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Estas funciones se encuentran en la biblioteca de funciones book_fne. php. El código de cada una de ellas se muestra en los listados 27.11 y 27.12 respectivamente.

Liatlldo 27.11. Funci6n calcula19_prtc:e() de book_fns.php. Esta funci6n calcula y devuelve el pNC:lo total de los contanidos del carro de la compra.

```
funccion calculace_price(Scarcl
(
   // calcule el precie cocal de Codoe loa arciculoa del carro
   Sprice • 0.0;
   1f(ia_array($carcl
   !
      Sconn • dll_connecc(l;
      foreach($carc •• $iai:D •> S~yl
      (
         Squery • 'aelecc price from booka vhere iabn•'$iabn'";
         Sreaulc • $co11D->queryl$query)1
         if (Sreaulcl
         (
            Sice• • Sreault->fecch_oojeetlli
            Sicem_price • $iC— ·>price1
            Sprice +•Sitem_price•$qcy;

      )
   ):
   recurn S,rice,
```

Como puede comprobar, el funcionamiento de calculate_price () consiste en buscar el precio de cada artículo en la base de datos. Es un poco lento. por lo que para evitar repetir esta operadón más de lo necesario, almacenaremos el precio (asf como el número tot~I de artículos) como variables de sesión y solamente repetiremos los c61culos cuando el carro cambie.

Listado 27.12. Funci6n calculate_items() de book_fns.php. Esta función calcula y devuelve el mlmero total de artlculos del carro de la compra.

```
funccion calculace_icema($carcl
(
   // •u.e todo• loa arciculoa del carro
   Sice- • 0;
   {f(ia_arr~y($cartll

      Sit•..•aZTay_aua($carc)1
   )
      require valid-user
```

La función calcuiate_iteme () es mú sencilla; simplemente suma las cantidades de cada artículo del carro para obtener el número total por atedio de la función array _eum(l. Si todavía no existe una matriz (si el carro está vado) devuelve 0.

Guardar el carro actualizado

Si hemos llegado a la secuencia de comandos sbov_cart .pbp tras pulsar el botón Save Chinga, el proceso es ligeramente distinto. En este caso, hemos accedido a trav• del envfo de un formulario. Si se fija en el código. comprobad que el botón Save Chinga es el botón de envfo de dicho formulario. Este formulario contiene la variable oculta save. Si configuramos esta variable, sabremos que hemos accedido a esta secuencia de comandos a través del botón Sa'ft Changa Esto significa que probablemente el usuario ha modificado los valores de cantidad del carro y que tendremos que actualizarlos. Si observa los cuadros de texto de la parte del formulario Save Clumps de la secuencia de comandos, comprobará que red·ben el nombre del ISBN del artfculoque representan,como se indica a continuacidn:

```
echo '<input type="text" name="$isbn" value="$qty" size="3">';
```

Ffjese ahora en la parte de la secuencia de comandos que guarda los cambios:

```
{f(iaaec($_POST['aave•Ji>

   foreach ($_SBSSION(•cart•J aa $ial:IID •> Sqtyl
   (
      1f($ l'OST[$1sbal• •'0')
         uiaetl$_SESSIOll['cart•J($1al:IIDJ);
      eln
         $_SISSIOll('cart•J [$uibDJ • $_POST[$1sbal,
   1
   S_SISSIOll['total.,JU"ice•J • calculate_price($_saSIOll['carc•J);
   $_SISSIOll['it...'] • calculate_1c...,t_SSSSIOll('cart'))1
```

Comprobad que estamos procesando el cam1de la compra y que, en cada iabn del carro, comprobamos la variable POST con ese nombre. Se trata de los campos del formulario Save Chulps. Si alguno de los campos se configura con el valor cero, eliminamos dicho artfculo del carro por medio deunset () . En caso contrario, actualizamosel carro para que coincida con los campos del formulario,de esta forma:

```
if($ POST[$1al:IIDJ• •'0')
   uiaet($_SISSION('cartJ [$iabnJl1
eln
   $_SBSSIOll('cart•] [Siabal • $_1'0ST($iabD)1
```

Despu• de realizar las actualizaciones, utilizamos calculate_price () y calculate_items () de nuevo para determinar los nuevos valores de las variables total_price e it ....

Imprimir un resumen en la barra de encabezado

Habd comprobado que en la barra de encabezado de todas las p4ginas del sitio se muestra un resumen del contenido del carro. Para obtener este resumeJ\, se·

EL_HOLISTICO@HOTMAIL.COM - 2006

imprimen los valores de las variables de sesión total_price e item,. operKión que se lleva a cabo en la función do_hmtl_header (>.

Estas variables se registran cuando el usuario visita por primera vez la página •how_cart .php. Tambim necesitaremos ciérta lógica para los casos"fll los que el usuario no haya visitado dicha p6gina. Esta lógica tambim se encuentra en la función do_html_header ():

```
if(IS 8&88IOll['it...•J) S 8&88IOll['it...J • 'D'1
if(IS:8888IOll('total_price•J) S_S&SSJOll('total_price•J • •0.00•1
```

Salir

Cuando el usuario pulsa el botón Go to Cbeckout desde el carro de la compra, se activa la secuencia de comandos checkout. php. A esta p6gina, as(como a las que se encuentran por detrú de la misma, se debe acceder a trav4s de SSL, pero la aplicación de ejemplo no le obliga a que lo haga (en capítulos anteriores encontrad mú información sobre SSL). La página de salida se muestra en la figura 27.8.

Figura 27.1. La HCUencla de comandos checkoulphp obtiene los detalles del cliente.

Esta secuc,ncia de comandos requiere que el cliente introduzca su dirección (y la dirección de envío en caso de que sea diferente). Es bastante sencilla, como puede comprobar en el c6digo del listado 27.13.

Llatado 27.13. c:hecllauLphp. Esta ~cill de CCIlMlldol oblIelle IDI ....._ del clienta.

```
c?pbp
   //incluya 111199tro coajuato de fuacioaee
   nquire  l•llook_ec_fae.php'l•

   // &l carro ele la compra aeceeita ..eioaee, inicie una
   ••••ioa_etartllr

   do_btal_beaderl•C1aeckout'll

   iflS_SBSSICll['cart•J•••rray_COUDt_valueelS_SISSICll['cart•J)l
   (
     dieplay_cartlS_S&SSIOlll'cart•J,  fal.., Dlr
     dieplay_cJwckout_fortl()l

   elee
     echo 'cp>Tber• an no it•-  in your cartc/p>' l

   dieplay_buttoal 'ebov_cart .pbp•, 'coatiaue-eboppiag',  'Continua Sboppiag' l r

   do_btal_footerll;
?>
```

No hay nada extrafto en esta secuencia de comandos. Si el carro est6 vado~ se lo indica al cliente; en caso contrario, muestra el formulario ~ la figura 27.8•

Si el usuario quiere continuar *y* pulsa el botón Pun:bue situado en la parte inferior del formulario, acceder6 a la secuencia de comandos purchaae . php, cuyo resultado se muestra en la figura 27.9.

El c6digo de esta secuencia de comandos es ligeramente mú complejo que el de checkout. php. Lo mostramos en el listado 27.14.

u.t.do 27.14. pun:hae.php. Ella secuencia de CIIIWldos.....,. los dellIlea del pedido en le bue de datos y obtiene los dellIlIN de pego.

```
c?pbp

   iaclude l'book_ec_fae.pbp'I,
   // El carro de la compra aeceeita ..eioaeer inicie una
   ••••ioa_etartll;

   do_btal_b9aderloCbeckout•l l

   //cree aoabree de variable• corto•
   Saa.. • S l'OST('aaae')r
   Saddreee:  S_POST['addreee•Jr
   Scity • S_POST['city•J;
   Saip • S_POIT['aip'lr
   Scouatry • S_POSTl'couatry•Jl

   // si se completa
   if(_SESSION['cart']&&$name&&$address&&$city&&$zip&&$country)
   {
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
;; .. puede aftadir a la l:laae da dato•
ifl  in•ert_ordertS_POSTl!•falH   I
(
   //~e•tre el carro,  •in permitir callbioe Y •in imtgen••
   di•play_cart(S_SISSIOJl(•cart'I,  fal•e,  01;

   di•play_•bippinglcalculata_•hipping_co•tCII;

   //o'otenga loe dato• da la tarjeta de crtdito
   di•play_card_fora(Sna..ll

   dieplay_buttonl'•bow_cart.pbp••  ·cont~-ebopping',•'coat1-    Sbapping•l,
1
el••
(
   echa •could not •tore data, pl•a••  try again.';
   di•play_buttonl'cbeckout.pbp',    'back',  •&ack'll


el••
(
   ecbo 'You did not fill in all  tbe fielda,  plea••  try again.cbr  />';
   dieplay_button('cbeckout.pbp',    'l:lack',   'Back'I;
   ~.
: do_buil_f—terll;
?>
```

Figura 27.9. La secuencia de comandoll pun:hase.pllp calcula el envfo y el total dal pedido. y obtiene los detalles de pago del diente.

La lógica es muy sencilla. ComF.()bamos que el usuario haya completado ei formulario e introduddo los detalles en la bue de datas p>l' medio de Ja función inHrt _order (›, Se trata de una función muy sencilla que aftade los detalles del cliente a la bue de datos. Su código se incluye en el listado 21.15.

u.t.clo 27.15. Función insert_order() de order_ma.php. Esta función allade todos loa detalles del pedido del cliente a la base da dalas.

```
function in•ert_ordar1$ordar_detail•l
~

  //extraiga  ordar.datail•  como variable•
  extractl$ordar_detail•I;

  //dafina la direcci6n da envio igual que la direc:ci&l normal
  ifl!$•bip_Dalle••1S•hip_addre••••'s•hip_city••
     !$•bip_•tate66!$ebip_zip66!$•hip_countryl

    $•bip na.. • Sname,
    S•bip=addr••• • $addr•••,
    S•hip_city  • $city,
    $ehip_•tate • $•tate,
    S•bip_zip  • $zipl
    $•bip_country • $couatry,


  ·—·    db_ ——— ll;

  // quere.o• alladir el pedido como tranaacci6n
  // inicie una da•activando la confiraaci6n automiltica
  Sconn->auiocmait IFALSll /

  //al'aada la direcci6ndel  cliente
  Squery • ''elect  cuatomerid froa cu•tomer•  where
            na.. • 'Snaae' and addr••• • 'Saddr•••'
            and city • '$city•  and •tate•  'S•tate•
            and zip • '$zip' and country •  '$Country••;
Sreeult • $CODD->queryl$query);
ifl$re•ult->num _row•>OI
(
   $cu.tomar•  $re•ult->fetch_o'ojectl);
   Scu•tomerid • $cu•to..r->eu•tomerid;


   $query • ''in..rt into cu.tomar• value•
         ("•    '$Da.e.,$addreaa•,'Scity','satate•,•$aip•,•$cOUDtry')•;
   Sr•eult  •  $CoDD->queryl$query);
   if   CISre•ult)
       return fal••;

   $customerid = $conn->insert_id;

   $date•   date('Y-•-d'J;
```

```
$query • "ia.ert iato order• walu••
      ,''' $cu•t01Mtrid, ''' _1saa1ox['total..,J>riee•J.•,    '$date•
         'PAJITIAL•  '•S•hip =  '
         '$•hip~addr••••,•i•hip_~ity','$•hi~  _•t•t•','S•hip_aip•,
         '$ahip_couatry•)~¡_        _

$reault • $coaa->quary($query)
if  (1$reault)
   retura fal•e;

$query • •aelect ordarid fr- ordar• whera
      cu•t01Mtrid • $cu•t01Mtrid aad
      a.ouat > ".$_SB81I01f[•total.:.Price•J.•-.001    aad
      AII0unt e ''' SBSSI01f['total..,J>ri~e·J.•+.001 aa4
      date•  '$datē' aad
      ordar_atatua • 'PARTIAL' aad
      •hip_aa_ •  '$ahip__   ' aad
      •hip_addre•• • '$•hip addre••' aad
      ahip_city • 'S•hip_city' aad
      •hip_atata • '$ahip_atata• aad
      ahip_aip • 'S•hip_aip• aad
      ahip_couatry • '$ahip_country''1
$re•ult • $cona->query($quary);
if($reault·>num _rowa>0I
{
   $order • $re•ult·>fatch object(li
   $order1d • $ordar->ordarid;

el•e
   ratura falaa;

// aliada lo• libroe
{oreach(S_SBSSI01f1'cart•J  H  $ieba o  Squantity)


$datail • tet~booll: detail•!Si•bal1
$query • "dalete fr-  ordar_itua  Whare
         orderid • '$orderid' aad ieba.  '$i•ba••1
Sr••ult • $coan->quary($query) ¡                  ·›_
$query • "iaaert into order_item• valu••
         ('$ordarid',   'Si•bn',  o.$data11[•prica•J   .',  $quaatity)•;
Sreault • Scoan->query(Squary) ¡                  .'
if(1$re•ult)
   retura falae,


// finalice la traa•acci6n
SC0IID->C01111it(11
Scoan->autoc0111lit(TRUS);

raturn $ordarid;
```

d La función es extensa ya que nos permite aftadir los detalles del diente los etalles del pedido y los detall~ de lodos los libros que quiere comprar.

Comprobar6 que las distintas partes de la operad6n se induyen en una transac-d6n. comenzando por

y tenninaJ.ufo con

```
$CODD•>C01 9it();
$C011D~>&UtOC01111it(TRUS)1
```

6- el 11nico punto de la aplicadcm en el que tenchemos que uWizar una transac-ci6n. Se preguntar, cómo evitamos tener que hacerlo en otros puntos. Ffjese en el c6digo de la función db_coanect ():

```
fuacti-  db_coaaect0
{
   Sra•ult •°new ayeqli('localhoat•,  'book_ac•,  •pa•eword',  'book_ac•>,
   U  (1$raault)
      ratura falaa;
   $raault•>AUtoc:01111it(TRUS)1
   ratura $raault.1
```

Evidentemente, es un tanto distinto al código que utilizamos en otros capftuUft para esta función. Tras crear la canexión a MySQL, debería activar el modo de confirmadón autom6tica.

De esta forma se asegura que todas las instrucciones SQL se ejecutan autom6ticamente, como ya hemos mendonado antes. Cuando desee utilizar una transacción de varias instrucciones, desactive la conlirmadón autom6tica, ejecute varias operadones de aftadir datos, conffrmelos y vuelva a activar la confirmaci6fl autom6tica.

Tras ello cargamos los gastos de envio a la ciln!cd6n del cliente y le indicamos a cumto ascienden por medio de la siguiente linea de código:

```
diaplay_•hippiag(calculata_ahippia9_coat(1l1
```

m c6digo que utilizamos para calculate _ahipping_coat (I siempre devuelve 20 dólares. Cuando disefte un sitio de compras, tendr6 que seleccionar un método de entrega, determinar los gastos en fund6n de los diferentes destinos y calcular el total correspondiente. Seguidamente, mostramos un formulario al usuario en el que debe introdudr los datos de su tarjeta de cridito, por medio de la función diaplay_card_form() de la biblioteca output_fna.php.

Implementar el pago

Cuando el usuario pulsa el bobSn **Pmdme,** procesaremos sus detalles de pago por medio de la lleCUenCia de comandos proceaa .php. En la figura 27:10 puede ver el resultado de un pago satisfactorio.

EL_HOLISTICO@HOTMAIL.COM - 2006

Flgul'll 27.1 D. Esta transac:clón es c:orrecta, por lo que M enviarán los artfculos.

En el listado 27.16se incluyeel c6digo de procesa. php.

Llahldo 27.11. process.php. Secuencia de comandos que procesa el pago del diente y le muestra el resultado.

```
c?php
   r.quire ('boolc_sc_fas.php'l;
   // Sl carro da la compra zwcesita aeaioaes; inicie una
   aeasioa_atartll,

   clo_htal_headar('Checlcout'I;

   Scard_typa • S_POST(•card_typa'll
   Scard_number • S_POST('card_nullbar'll
   Scard_acmth • S_POST(•card_month• I;
   Scard_yaar • S_POST('card_year'l;
   Scard_na_ • S_POST(•card_— ·1;

   ifCS_SISSlOJl[•cart•J••Scard_type66Scard_number..
       Scard_acmthAAScard_yaar••scard_namal

      //.ueatre el carro, aia permitir cllllbioa y aia im&ganea
      diaplay_ca~CS_SESSlOll('cart'l, false, OI;

      display_ahipp1agCcalculata_ahipping_costlll;

      iflproceaa_cardCS_POSTII
```



```
      //vacie el carro da la coapra
      aeaaion_daatroyll;
      echo 'Thanlcyou for ahopping vith ua. Your ordar haa baan placed.' l
      display_button('index.php', 'c:aitillue-ahopping', 'Cllllltimue Sbopping' I;

   elae
   (
   ecbo 'could not procesa your card. 'l
   echo 'Please contact the card iaauer or try again.';
      diaplay_button(•purchase.php', 'baclc', •laclc'll

]
elae
(
   echo 'You did not fill in all the fields, pleaae try again.chr />';
   diaplay_buttoal'purcha ... php', 'baclc', '&acl•');


do_htlll_footer(I;
```

Procesamos la tarjeta del usuario y si todo es correcto, destnaiinos su sesión. La función de procesamiento de la tarjeta, como la hemoe escrito, simplemente devuelve true. Si tuviéramos que implementarlo realmente, necesitaríamos algdn tipo de validación (comprobar que la fecha de caduddad es válida así como el número de la tarjeta de crédito) y, tras ello, procesar el pago.

Al disei\ar un sitio real, tendrá que tomar una decisión sobre qué mecanismo de compensación de transaccionesva a utilizar. Dispone de diferentes opciones:

- Puede contratarun servicio de compensación de transaccionesde un proveedor. En función de la zona en la que viva, encontrará gran cantidad de alternativas. Algunas le ofrecen compensación en tiempo real y otras no. Si tiene pensado ofrecer un servicio en linea, le resultará imprescindible; si solamente tiene pensado enviar productos, será menos importante. De cualquier forma, estos proveedores evitan tener que almacenar números de tarjetas de crédito.
- Puede enviarsea sí mismo un número de tarjeta de crédito a través de correo electrónico codificado, por ejemplo, con ayuda de PCP o GPG, como vimos en un capftulo anterior. Cuando reciba y descodifique el correo, podrá procesar las transacciones manualmente.
- Puede almacenar los números de las tarjetas de crédito en la base de datos. Esta opción no es recomendable a menos que sepa realmente lo que hace con la seguridad del sistema. En capítulos anteriores enc:ontrar6 más.infimnac:ión al respecto.

Esto es todo lo ql:le necesitamos para los módulos del carro de la compra y de pago.

EL_HOLISTICO@HOTMAIL.COM - 2006

Implementar una interfaz de administración

La inteñaz de administración que hemos.implementado es muy sencilla. Simplemente hemos disellado una interfaz Web a la base de datos con un sistema de autenticación cliente. Es muy similar al código utilizado en uno de los capftulos anteriores. Lo hemos incluido para que la descripción resulte mú completa pero no nos detendremos en su análisis. La interfaz de administración requiere que el usuario se conecte por medio del archivo login. php, que le lleva al menú de administración, admin. php. Puede ver la página de conexión en la figura 27.11 (hemos omitido el archivo login. php por motivos de brevedad, aunque es prácticamente idéntico al del capftulo anterior; no obstante, se incluye en el CD-ROM). El menú de administración se muestra en la figura 27.12.

Figura 27.11. El usuario daba pasar por la p4glna da Inicio da sasi6n para acceder a las funcionas da administración.

En el listado 27.17 se recoge el código del menú de administración.

Ustado 27.17. admln.php. Esta secuencia da comandos autentica al administrador y la parmlta acceder a las funcionas da admlnlstracl6n.

```
c?php

// incluya loa archivo• de incluai6n de eata aplicaci6n
require_once('book_ac_fna.php'I;
••••ion_•tart(I;
```



```
// acaban de intentar conectar••
(

   $uaername • $ POST[•u..rname•J;
   $paH~ _• $,,,~['paaawd');

   if  (login($uHmuil,  $paHwd) l
   (
     //  •i  eatan en la baH de datoa,  regiatre el Id.  de uauerio
     $_SUSIOR(•adllin_ÚHr'l  •  $uaerna •;
   }
   el••
   (
     // conexi6n fallide
     do btal header(•Probl..:•)1
     echo 'Y~u could not be logged in.
            You mu•t be logged in to view thb  page,cbr />';
     do btml url(•login.php•, •Login•);
     do:btal:footerll1
     exitl


do btal beader('Adllini•tration•I,
if-(cbeck_aclmin_uaer(II
   diaplay_adain_11eDu(l1
el••
   echo 'You are not autborised to enter tbe adainiatration area.•;

do_html_footer();
```

Figura 27.12. 8 manll de administración permite accader a las funciones de administración.

EL_HOLISTICO@HOTMAIL.COM - 2006

Probablemente este c:ódigo le resulte familiar, ya que es parecido a una secuencia de comandos del capitulo anterior. Cuando el administrador llega hasta este punto, puede cambiar su contraseña o desc:onec:tane. Como el c:ódigoes idéntico al del capítulo anterior, no lo describiremos en este apartado.

Una vez conectado, identificamos al usuario administrador por medio de la variable de sesión admiu_ uHr y con la función check_adllin_user ( ) . Esta fun. ció6n, así como todas las utilizadas por las secuencias de comandos administrativas, se encuentra en la biblioteca de funciones admin_fna. php.

Si el administrador opta por añadir una nueva categoría o un libro, puede hacerlo desde inaert_category_form.php o inaert_book_form.php, respectivamente. Cada una de estas secuencias de comandos presenfa al administrador un formulario que debe rellenar. Cada uno se procesa por la correspondiente secuencia de comandos (inaert_category_form.php e inaert_book_forlll.pbp), que verifica que el formulario se ha rellenado y ai\ade los nuevos datos a la base de datos. Solamente nos detendremos en la secuencia de comandos para libros, ya que ambas son muy similares. El resultado de iuaert_book_form. php se muestra en la figura 27.13.

Flguni 'D.13. Esf8 formul.io permite al adminillrador aftadlr nuevos llbloa al catMogo en linea.

Habr6 apreciado que el campo Category para los libros es un elemento SBLBCT de HTML Las opciones de SBLBCT provienen de una llamada a la función get_categori•• () que desaibimos anteriormente.

Al pulsar el bot6nAddBook. se activa la secuencia de c:mnandosinaert_book. php, c:uyo c:ódigo mostramos en el listado 27.18.

Listado 'D.11. inlett_boolc.php. Esta secuencie de c:omMdos Yllllda los datos de los nuevos libros y los aftade a la bMe de delos.

```
c?php

// incluya.lo• archivo• da inclu•ill!D de aata aplicaaiil!D
require_oacal'book_ac_fna.php'l1
••••ioa_•tartl1,

do_html_headllr('Adding a book'I;
if (check_adllinu ..r(II
(
   if (fillad_outl$_POSTII
   (
      $i•bD • $_POST['i•bn');
      $title • S_POST['title•J;
      $author • i POST['author'J;
      $catid • $ POST['catid');
      $prica • s:POST[•price•J;
      Sde•criptioD • $_POST['de•cription•);

      if(in•ert_book($iabn, $title, $autllor, Scatid; Sprice, Sde•criptionll
      echo ºBook ••.•trip•la•h••IStitlel .•• wa• added to thll databa•a.cbr />"l
      el••
         echo •aook ••.strip•l••ha•IStitla).
               •• could not be addad to the databama.cbr />";

   else
      echo 'You bava not fÚled 'out tha form. Plaa .. try a9ain.';
   do_btal_url('adain.php', 'Back to adllini•tration ..au•);

el••
   echo 'You ara not authoriaad to view thi• psp. ';

do_html_footar(I;

?>
```

Comprobar, que esta sec:uenc:ia de comandos invoca la función insert _book <) • Esta función, así como las funciones restantes utilizadas por las secuencias de comandos administrativas, se almac:enaen la bibliotecade func:ionesadlaiD_fus • php.

Ademú de poder ai\adir nuevas categorias y libros, el usuario administrativo puede modificar y eliminar estos elementos. Hemos implementado estas opciones reutilizando la mayor cantidad de c:ódigoposible. Cuando el administrador pulsa sobre el enlace Go to Main Slte en el menú de administración, accede al índice de categorías de inclex. php y puede navegar por el sitio igual que un us\18rio convencional. utilizando las mismas secuencias de comandos.

Sin embargo, ~te una diferencia con la navegación de administración. 1..m: administradores venn distintas opciones ya que disponen de la variable de sesión~

EL_HOLISTICO@HOTMAIL.COM - 2006

registrada admin_user. Por ejemplo, si se fija en la página •how_book. php anterior, verá que hay opciones de menó diferentes. P(jese en la figura 27.14.

Figura 27.14. La secuencia de comandos ahow_book.php genera resultados diferente, para loa uauarlol adminiatrallvol.

En esta página, el administrador tiene acceso a dos nuevas opciones: &lit Item y Admin Menu. También comprobará que en la esquina superior derecha ya no se incluye el carro de la compra sino un botón Log Oat. Todo este código se concentra en el siguiente fragmento, tomado del listado 27.8: ...

```
if( check_admin_uaerll  l
{
|t-diaplay_buttoa(•edit_book_fono.pbp?iaba•$iabn•, 'edit·it ..•, 'ldit
   ,,
   diaplay_button(•admia.pbp',   •admia·menu•, 'Adllin Neau'),
   diaplay_button($target,  'continua•,  'continue'l1
}
```

Si se fija de nuevo en la secuencia de comandos ahow_cat.php, comprobard que también cuenta con tres opciones. Si el administrador pulsa el botón &lit Item, accederá a la secuencia de comandos edit_book_form.php, cuyo resultado se muestra en la figura 27.15. Se trata, de hecho, del mismo formulario que utilizamos anteriormente para obtener los detalles de los libros. En dicho formulario hemos incluido una opción para pasar y mostrar los datos de los libros existentes. Hicimos lo mismo con el formulario de ~tegorfas. Lo entenderd mejor con ayuda del listado 27.19.

Figura rr.15. La secuencia de comandos edlt_book_form.php pennlte al administrador modificar loa detalles de un libro o ellmlnarto •

Uatado 27.11. Función dlaplay_book_form() de admln_fns.php. Este formulario funciona tanto como formularlo da lnsen:l6n como da modlflcacl6n.

```
function diaplay_book_form($book • ''l
// Mueatra el fozmulario de libroa.
// la lll1y aiailar al fozmulario de categorlaa.
// Se puede utiliaar para aftadir o II0dificar libroa.
// Para •lladir libroa, no paae ningGn para-tro. Configurara $edit-
// con el valor falae, y el formulario ira llaata inaert_book.php.
// Para actualiaar libroa, paae una ..tria que contenga el libro. 11
// formulario u — trara coa loa datoa antiguoa y apuntara• update_book.pbp.
// TallbHn aftada un bot6n •Delete book•.
(

   // ai •• ba paaado un libro exiatente, proaiga en •modo edici6n•
   $edit • ia_array($book);

   // la aayor parte del fonoulario eata en ll'nlL aiaple con algunoa
   // fragaeatoa de PHP opcioaalea

   cform metbod••poat•
     action••c7pbp echo $edit?•edit_book.pbp':'inaert_book.pbp'1?>">
   ctable border•"0">
   ctr>
     ctd>IIBll:c/td>
     ctd>cinput type••text•  na..••iebn•
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
        valae••c?pbp acbo Sadit?Sllook[•iabn%:''1    ?>•>c/td>
    c/tr>
    ctr>
       ctd>look Titla,c/td>
       ctd>eiaput typa••text• ......•titla•
          value••c?pbp acbo Sadit?Sllook['C.itla'l•''1   ?>•>e/%d>
    c/tr>
    ctr>
       ctd>look Autbor:c/td>
       ctd>ciapat typa••taxt• ......•autbor•
            valua•°c?pbp acbo Sadit?Sllook['autbor•1,••1    ?>°>c/td>
     c/tr>
     ctr>
       ctd>Catagory,cltd>
       ctd>caalect ......•catid°>
       c?pbp
          // la liata da poaiblaa catagorlaa •• obtiene da la baaa da datoa
          Scat_array•gat_catagoriaall1
          foraacb (Scat_array .. Stbiacatl
             acbo •coptioa value•••,
             acbo Stbiacat(•catid'l1
             echo••';
             // ai •• un libro axiataata, ul&dalo a la catagorla actual
             if ISadit •• Stbiacat(•catid'I      $book(•catid'II
                acbo ' aalactad'1
             ecbo »•;
             acbo Stbiacat[•cat .....'];
             ac:bo •c\optioa\a•;
          c/aalect>
       c/td>
    c/tr>
    •tr>
       c'C.d>Prica,c/'%d>
       c'%d>ciapu%  typa••taxt•  .....•prica•
                    valua••c?pbp acbo Sadit?Sbook(•prica•] ,••;   ?>°>c/td>
    c/tr>
    ctr>
        ctd>Daacriptioa,c/td>
        ctd>ctextaraa ro,ni.•3• col•••so•
          .....•daacriptioa°>c?pbp
          acbo Sadit?Sbook('daacriptioa'):''1 ?>c/textaraa>c/td>
    c/%r>
    cCr•
       ce• c>pbp if (ISaditl acbo 'colapaa•"2"'; ?> aliga.•caatar•>
                U ISaclitl
                // nacaalt._a al ISU antiguo para localisar al libro ea la
                //Naa da datoa
                // •l al Is .. •• actualisa
                acu 'c111p111: typa••biddaa• —  - ·oldialm•
                   val11a.••.Sbook['iaba•J.•o>'1
        ciapl1C 11:)'pa•°aubmit•
```



```
        valua•°c?pbp acbo $adit?'1Jpdata•:•Add•;   ?> 8ook">
    c/fora>c/td>
    c?pbp
       if (Sadit1
       (
          ecbo •ctd>',
          acbo 'cfora ..tbad-°poat• actiaa•"dalata_book.pbp°>';
          acbo 'cinput typa••biddaa• .....•iaba•
                valua•°' .Sbook[•i.—•J. '°>' 1
          acbo 'ciaput typa•°aubmit•
                valua•"Dalata book">';
          acbo •c/fora>c/td>'1


        c/r:.d>
    c/r:.r>
  e/tabla>
  c/fora>
c?pbp
}
```

Si pasamos una matriz que contenga los datos de los libros, el formulario ie representad en modo de edición y completad los campos con los datos existentes:

```
ciaput typa••taxt• aa..••prica•
        valua••c?pbp acbo $adit?$book['prica•1,''1    ?>°>
```

!memos .ioduao un botón de fflY(o diferente. De htcbo, para eJJg~ulario de edición tenemos dos: uno para actualizar el libro y otro para eliminarlo. Con atoe se invocan las secuencias de comandos edit_book.php y delete_book.php. que actualizan convenientemente la bue de datos. Las versiones.decategorías.de.... estas secuencias de comandos funcionan de forma similar a excepción de un aspecto. Cuando un administrador intenta eliminar una categorfa, no la podr6 eliminar si contiene algún libro (lo que se puede comprobar por medio de una ~ulta a la base de datos). De esta forma se evitan todos los problemas que podamos tener con anoma!fas de eliminación, como mencionamos en uoo de los primeros capitaloe. En este caso, si se elimina una categoría que incluya libros, átos quedarfan hu4rfanos• No podremos saber en qu4 categoría se encuentran y no podremos desplazarnos hasta ellos. Con esto hemos terminado el repaso a la interfaz de usuario. Si necesita mú información, puede consultar el código incluido en el CD-ROM •

Ampliar el proyecto

Hemos construido un sistema de carro de la compra bastante sencillo. Sin embargo, podemos realizar numerosas mejoras al mismo:

- En una tienda en linea real, necesitaríamos algún tipo de sistema seguimiento y recepción de pedidos. Por el momento, no hay forma de saber qui pedidos se han realizado.

EL_HOLISTICO@HOTMAIL.COM - 2006

- Loe clientes deben poder comprobar el progreso de sus pedidos sin -- que ponene en contacto con noeotroe. Es importante que el cliente no tenga que conectane pan navegar. Sin embargo. si propord0Nlll08 a loa clientes existentes una forma de autenticarle, podón ver pedidos realizados previamente y nos permitid combinar C0111portamientos pan crear un perfil.
- Por ahora, es necesario enviar por FI'P las imigenes de los libros al directorio de ilúgenes y uignarles el~ c:orncto. Poddamoe cargar archivos a la p4gina de inlerc:i6n de libros pan que resultara 11"9 sencillo.
- Podríamos ailadir inicio de eesión de usuarios, penonalizad6n y recomendaciones de libros, crfticu en linea, programas afiliados, comprobad6n de las existencias, etc. Las posibilidades son infinitas.

Utilizar un sistema existente

Si quiere IUI carro de la compra totalmente equipado y operativo, puede optar por utilizar un sistema de carro de la compra existente. Un carro de c6digo abierto muy conocido que se implementa en PHP es FishCartSQL, que puede encontrar en http://www.fi8hcart.org.

Düpone de un gran nl1mero de opciones avanzadas como el seguimiento de pientes, ventas cronometradas,varios idiomu, procesamiento de tarjetas de c:Ñdito y compatibilidad con varias tiendas en linea en un aervidor. Evidenlemente, al utilizar un sistema ya existente, encontrar6 que no dispone de algunos elementos que necesita y vicevena. La ventaja de un producto de c6digo abierto es que aiempre se puedan cambiar los aspectos que no nos convenzan.

A continuación

En el siguiente capftulo aprenderemos a generar un sistema de gestión de contenidos en linea~muy indicado pan administrar activos digitales, lo que puede resultar de gran utilidad si contamos con un sitio buado en contenidos.

EL_HOLISTICO@HOTMAIL.COM - 2006

28

Crear un sistema de administración de contenidos

En este capítulo crearemos un sistema de administración de contenidos que nos permitirá almacenar, indexar y buscar textos y contenidos multimedia. Los sistemas de administración de contenidos resultan extremadamente útiles en sitios Web en los que el contenido del mismo lo mantiene más de un autor, en los que el mantenimiento lo llevan a cabo empleados que no son técnicos o en los que el contenido y el diseño gráfico es competencia de distintos departamentos o diferentes personas.

Generaremos una aplicación que permita a los usuarios autorizados gestionar los activos digitales de una organización.

Nos centraremos en los siguientes aspectos:

- Presentar páginas Web por medio de diferentes plantillas
- Crear un motor de búsqueda que indexe documentos en función de metadatos

El problema

Imagine que el equipo de desarrollo Web de SuperFastOnlineNews está formado por un excelente diseñador gráfico y por un determinado número de galardonados escritores. El sitio cuenta con páginas de noticias, de deportes y de información meteorológica que se ~ctualizan regularmente. La página principal muestra los últimos titulares de las tres páginas de categorías anteriores.

EL_HOLISTICO@HOTMAIL.COM - 2006

En SuperFastOnlineNews, los diseñadorepgarantizan el aspecto perfecto.de los contenidos del sitio Web. Es lo que mejor saben hacer. Por su parte, los escritores se encargan de redactar los mejores artículos pero no saben dibujar bien ni generar sitios Web.

Tendremos que permitir que cada uno se centre eri lo que mejor sabe hacer y que todos compartan sus resultad~ para de esta forma ofrecer el servicio superr.6pido de noticias que anuncia-el sitio.

Requisitos de la solución

Nuestro objetivo ser.6diseftar un sistema que:

- Aumente la productividad siempre que los escritores se centren en escribir y los disei\adores en diseñar.
- Permita al editor revisar los artfculos y decidir cu.61es se van a publicar.
- Cuente con un aspecto visual y operativo coherente en todo el sitio, obtenido por medio dt plan~las de p.6ginas.
- Permita que los escritores puedan acceder únicamente a las zonas del sitio asignadas para ellos.
- Permita que se pueda cambiar el aspecto visual y operativo de una determinada sección o de todo el sitio.
- Evite que se cambie el contenidoen directo.

Sistemas existentes

Existen numerosos sistemas de administración df' contenido,s, tanto gratuitos como comerciales. Antes de diseftar el suyo propio, puede que le interese examinar otros. Las ventajas e inconvenientes entre utilizar un sistema de otros y escribir uno propio son similares a las de otros proyectos.

La aeación de un sistema propio ofrece una gran flexibilidad pero tambicfn requiere mú trabajo. Puede decidir con exactitud cómo se integrarm los resultados en el sitio Web y cómo se procesar-' el contenido dinúnico.

Los sistemas existentes pueden proporcionarle características avanzadas sin apenas trabajo. Suelen generar resultados muy flexibles ya que es uno de los objetivos de estos sistemas pero nos limitan a un determinado flujo de trabajo y no procesan el contenido din.6mico demasiado bien.

El disei\o de un sis~ma sencillo, como las caracterfsticas que crearemos en este capítulo, no lleva mucho tiempo, pero cuantas mú opciones necesite mayor ser.6 la dificultad y mú atractiva se.hace la posibilidad de adoptar un sistema ya existente.

Modificar el contenido

Eñ primer lugar, tendremos que pensar en c6mo aftadlremos el contenido al ~yen amo lo almacenaremos y moc:U&camnaa.

Añ1dircontenido en el sistema

Tenchemosque decidir qucf artfculos y componentes de cilaeilo se enviarm. Existen varias posibilidades, aunque en los siguientes aputados sólo analizaremos tres de ellas.

FTP/SCP

Los eecritons y disei\adores pueden tener acceso a traws de Fl'P o 5CP a zonas del servidor Web en las que puedan cargar archivos desde su equipo local. Necesitaremos un estúldar de notación fija para los archivos cargados (para identificar qué im.6genes corresponden a cada artículo) o un sistema basado en la Web para hacerlo de forma separada a la carga desde Fl'P. Desafortunadamente, puede resultar complicado conceder derechos y permisos a distintos usuarios por medio de estos mcftodos, por lo no lo utilizaremos en este ejemplo.

Método de carga de archivos

Como ~ en un capítulo anterior, el protocolo H'ITP cuenta con un método para cargar archivos por medio del navegador Web. PHP lo puede procesar de forma muy sencilla.

El método de carga de archivos tambicfn nos permite almacenar texto en una base de datos en lugar de hacerlo en forma de archivo. El mecanismo de carga crea un archivo temporal. Para almacenar los datos en un archivo, se copian a una ubicación permanente. Para almacenarlos en la base ~ datos, podemos leer en el archivo temporal y almacenar sus contenidos en la base de datos. Opcionalmente podríamos utilizar la carga de archivos para loe articulos de este proyecto pero lo necesitaremos para las imgenes.

Cambios en línea

Podemos permitir que los usuarios creen y modifiquen documentos sin necesidad de utilizar FTP,CSP o la carga de archivos. Por el contrario, podemos ofrecer a los colaboradons un cuadro de entrada de texto en pantalla en el que puedan modificar el contenido de sus artículos.

Este método es muy sencillo, pero eficaz. El navegador Web no cuenta con ninguna función de edición de texto aparte de las opciones de cortar y pegar del sistema operativo. Sin embargo, cuando es necesario realizar algún cambio, por ejemplo, corregir un error ortoggfico, la operación de seleccionar el contenido y corregirlo es muy r.6pida. Desafortunadamente, como un elemento textarea de

EL_HOLISTICO@HOTMAIL.COM - 2006

tITML no cuenta con opciones avanzadas como la revisión ortográfica en tiempo real, es ms que probable que tenga que corregir personalmente los errores.

Al igual que la carga de archivos, los datos del formulario se pueden escribir en un archivo o se pueden almacenar en una base de datos.

Bases de datos frente a almacenamiento en archivos

Una de las decisiones más importantes que debemos tomar en las fases iniciales es determinar cómo se almacenará el contenido una vez cargado al sistema.

Como almacenaremos metadatos junto al texto del artículo: hemos optado por aftadir las partes de texto del contenido a la base de datos. Aunque MySQL puede almacenar datos multimedia, optamos por almacenar imágenes cargadas en el sistema de archivos. Como mencionamos en capítulos anteriores, el uso de datos BLOB en una base datos MySQL puede reducir el rendimiento.

Simplemente almacenaremos el nombre de archivo de la imagen en la base de datos. Al utilizar el sistema de archivos, la etiqueta <IMG> puede hacer referencia directamente al archivo de imagen, como de costumbre.

Estructura de documentos

tos. artículos de ejemplo que utilizaremos son breves, de uno o dos párrafos con una sola imasen, diseftados para usuarios con prisa. Se trata de documentos estructurados ya que contienen un titular y uno o dos párrafos de texto con una imagen. Cuanto mayor sea la estructura del documento, con mayor facilidad se podrá dividir para almacenarlo en una base de datos. La ventaja es que todos los documentos se pueden presentar de forma estructurada y coherente. El principal inconveniente es que a ms estructura menos flexibilidad.

Veamos nuestro ejemplo de artículos de noticias. Almacenaremos el titular en un campo aparte del texto del artículo y, por naturaleza, la imagen es un componente independiente del documento.

Con el titular como elemento independiente, podemos definir un tipo y un estilo estándar con el que lo representaremos y podremos separarlo del resto del artículo para crear nuestra página de titulares principal. Otro enfoque que podríamos utilizar en documentos de gran tamaño consiste en aplicar una relación uno a varios entre los distintos párrafos, es decir, almacenar cada párrafo en una fila de la base de datos y vincular cada uno a un Id. del documento principal Este tipo de estructura dinámica de documentos nos proporcionaría una mayor flexibilidad en lo que respecta a la estructura y representación de los documentos.

Utilizar metadatos

Ya hemos decidido que cada registro de articulo incluin un titular, el texto del articulo y una imagen. Sin embargo, podemos almacenar otros datos en el mismo

registro. De fonna automtica, nuestro sistema añadirá valores para el que haya creado el articulo y para la última fecha en la que se haya modificado. Estos valores se pueden representar en la parte inferior del articulo para añadir una fecha y una marca de tiempo sin que el autor tenga que preocuparse de incluir esta información. También puede resultar útil añadir datos que no se muestren, lo que se denomina metadatos. Un buen ejemplo seria almacenar palabras clave que se podrían utilizar en el índice del motor de búsqueda.

En lugar de tener que buscar por todo el texto de todos los artfculos, el motor de búsqueda se centrará en los metadatos de palabra clave de cada articulo para determinar la importancia a partir de dichos datos. Esto permite que el administrador del sitio disponga de control total sobre qué palabras y frases de búsqueda coinciden con qué documentos.

En nuestro ejemplo, permitiremos que se pueda asociar cualquier número de palabras clave a un articulo y asignaremos a cada palabra clave un valor que indique la relevancia de la misma en una escala del 1 al 10.

De esta forma, podemos diseñar un algoritmo de motor de búsqueda que clasüique las coincidencias en los artfculos en funddn de esta relevancia, en lugar de un complejo algoritmo que tenga que interpretar el idioma y tomar decisiones en función de sus limitados conocimientos y las reglas fijas que lo controlan.

Si los datos son exclusivamente texto almacenado en una base de datos, la indización y búsqueda de texto de MySQL sería una mejor solución que desarrollar un sistema propio, pero el de nuestro ejemplo se podría utilizar como motor de búsqueda entre distintos tipos de documentos y archivos multimedia. Esto no significa que tengamos que almacenar los metadatos en la base de datos. Nada le impide utilizar la etiqueta <MBTA> en HTML o incluso recurrir a XML para generar los documentos. Sin embargo, siempre que sea posible, merece la pena aprovechar las ventajas de la base de datos en la que ya se encuentran los documentos.

Aplicar formato al resultado

A la hora de representar una página, nuestro sitio de noticias utiliza un sencillo pero estructurado formato. Cada página contiene un número de artfculos, tocios con el mismo formato. En primer lugar, se muestra el titular con letras grandes, seguido por la fotografía, situada más abajo y a la izquierda y, por último, el texto del articulo a la derecha. Toda la página se incluye en una plantilla de página estándar para garantizar el estilo y la coherencia de todo el sitio.

En la figura 28.1 podemos ver la estructura lógica de páginas que vamos a utilizar.

La implementación de una estructura de plantilla como ésta a partir de un disefto de página es muy sencilla. Simplemente se divide la página en tres partes: un encabezado, un menú lateral y un pie de página que no cambiall, y el contenido que varía diariamente de '!Jll8 página a otra. Siempre que se represente una página, primero mostraremos el encabezado, luego la página y por último el pie.

EL_HOLISTICO@HOTMAIL.COM - 2006

FIgIII'II 21.1. EltnIclura lógica de ptglna.

El encabezado de p6gina y el menl1 lateral se encuentran en un archivo (header. php)y el pie de p6gina en otro (footer. php). El contenido principal de cada p4gina se genera mediante la correspondiente secuencia de comandos.

La implementación de un sitio con una plantilla de encabezado y de pie de p6gina permite que estos archivos de plantilla se puedan cambiar con facilidad si se actualiza el disefto del sitio.

Diseñar y presentar la solución

En la tabla 28.1 se recoge un resumen de los archivos de esta aplicación.

Tabla 21.1. An:hlvoe de la apllcacl6n de admlnlatracl6n de contenido.

include_fllll.pbp	F~	Coleccl6n de lfdlivwde lnc:lU816ncle mplac:l6n
db_fne.pbp	F~	Coleccl6n ele func:loMstm11 canect8rM a la bme ele . . . de contallldol
Hlect_f1111.pbp	~	Coleccl6llderuncior. .,....cns.....SIILIIO'
u..r_auth_fna.pbp	~	Colllcci6nelefunclanN aullnllc:ar .....
header.pbp	Planlla	Se m-lnl en la parte 1Uperlorde locla1._ ptgln• da con1811idoa
footer.php	Planma	Se-11n1 en la perte lnfarlorde fDdal ._ pégln • de COllellidol
logo.9if	Imagan	ArdIlvoda logollpo IIIDIlrado anbeader. pbp
index.pbp	Aplk:aci6II	Rauman qua mua11n1al lllula1 mil raclanle de cada ptglliadallllo

class classname
{

adaln/iadex.pbp	Aplll:acl6n	Man6dalalfunclonft.....,.._clellllo.
pqe.pbp	AfllcaCl6II	E— . lollllularNyellaxlDdalol . . . . __da_ .... ptglIia
re•l••-1~.pbp	Aplll:acl6n	camlllaelllmaftOda — —— utnll ..... lludlinea.pbp
•••rcb_fora.pbp	Aplll:acl6n	F--1o para lnlraduc:lrpalabrNdlva .... -- por llll OIIIIIIIilldol del lllo
Harcb.pbp	Aplll:acl6n	.... da c:onlanldo qua --- GIIII llll crbrtosda..... clave
login.pbp	Aplll:acl6n	Alllanllc81a.....,..da un . . . . . ,..,1acanac1aa1 •••••••
logout.pbp	Aplll:acl6n	oa-iactaal ulUallodal sIItemll
wrlter.pbp	Aplll:acl6n	LIIta ele artlculol qua al UIUarlo conamdo ha .atto --opclOnperañdlr,moclllcaro..,._rllcukll
.cory.pbp	Aplll:acl6n	P..... dedaldNdeartlculalperamodlllcalOalladlr un..vo artlculo
atory_eubllll.pbp	Aplll:acl6n	Allllda —— artlculoloapllc:81111C8fflblolapertlrda IDININ1nlracluc:ldNenatory. pbp
delete~atory.pbp	Apllc:acl6rl	P — lalOllclluddaellmlnad6ndaunarlculodaacle atorlea.pbp
keyworda.pbp	Apllc:acl6rl	Enumaia • palaln1 clave da un articulo GIIII una opcl6npeiaalladlroallmlnarpelallraldlva
keyword_ad&S.pbp	Aplll:acl6n	Promaa la IIOllc:llud da alladlr palabras clave da keyworda.pbp
keyword_delete.pbp	Aplll:acl6n	Promaa .. IOlk:llud ele ellmlnar ..... clave ele keyworda.pbp
publiah.pbp	Apllc:acl6rl	LIalll ele artlculol del adllor en elqua H """*8111111 pijlllcadol con- opcl6n para CIIIIIIIIarel astado de lllclaellol
p~liab_atory.pbp	Aplll:acl6n	,- unalOllc:IIUd ele publlc:8d6ndepubliab. pbp
unp~liab_atory.pbp	Aplll:acl6n	Promaa unalOllcllud danop11bllcec:i6n elepubl iab. pbp

Diseñar la base de datos

En el listado 28.1 se incluyen las consultas SQL para crear la base de datos a partir del sistema de contenidos. _Este listado forma parte d~l archivo create_ d&tabaae.aql. El archivo indwdo en el CD también contiene consultas para aftadir usuarios y artfculos de e¡emplo a la base de datos.

u..- 21.1.Ї....., de awme_d....._ 1q1. An:Nva SQL.,.. definir i. bue de ctam eta c::ant•nldoa.

```
crNte acaban coatnt1

uee contea.e,

drop table if exiat• vriter•1

create tabl• writar• 1
   ueern-    varcharll&l  aot null primary Jcay,
   P•••-:rd  varchar1401  aot null,
   full_nama  tut
);

drop  tabla  if axi•t•  •toriee1

craate tabla etori•• 1
   id          int primary keyauto  incr...nt,
   writar      varcharll"  not null7  1l claft eecundaria writere.unrn-
   lMII•,:     varcharlHI  not null,     1 clave eecundaria page•.code
   beacl1ine   tut,
   etory_tut   tut,
   picture     tut,
   cnatad      int,
   madified    int,
   pOIIJlebed  1".
)1

drop ~~ble_ ~l-~•t•.PAP•!

craata table PA99• 1
   c:ode  varcber!H)  pri-ry key,
   de•criptiontut
)1

drop  table  if  ezi•t•  vritar_permieeiona1

craata table writar_perlli••ion• (
   writer    varcllarll&)  not -11.   1 cla,,. eecundaria writere.ueer-
   PA99      varcllarlHI  :mt  null,    1 claft eecundaria pap•.code
   primary kay(writer, ¡;ave)
11

drop  table if ezi•t•  keyworde1

create table keyo,ord8 1
   etory      int a.oc:  mal 1,     1 cla,,. IMCllldaria •torie•. id
   lcer-zd    varcbarll2t  not  null,
   -iglat     int not aall
   pri-ry     kayCetory, Jc~l
)1

vr-t  •elect, ineert, upckte, delete
```



```
on content.•
to conteatelocalbo•t identified by  •pu•II0:rd'1
```

En la tabla writers tendremos que almacenar derta información sobre cada uno de los escritores, incluyendo un nombre de inicio de sesión y una contrueila. Almacenaremos sus nombres completos para ailadirlos en cada artículo y para saludarloe cuando se conec:ten.

La tabla pagH contiene el encabezado de cada página en el que se muestran los utfculos. La tabla wri ter _peraisaiona implementa una relaci6n varios a varios que indica a qu! páginas puede un escritor enviar artfculos. La tabla stories contiene campoe independientes para headline, story _text y picture, como indicamos anteriormente. Los camposcreated, modified ypubliahed son campos enteros que almacenar6n el valor de la marca de tiempo de Unix cuando sea necesario. Para crear la base de datos, e¡ecute el siguiente comando:

```
-,.ql -u root e craate_databaee.eql
```

Asegúrese de que no cuenta ya con la base de datos content ya que de ser uf se eliminarfa y se sustituiría por áta.

Implementar el sistema

Una vn que e.n.noe la bue de datos y unafund6n para.cambiar el tamai\o de las ilúgenes, pueremos a crear la parte prindpal del sistema.

Interfaz de usuario

En primer lugar nos detendremos en iD.dex. php (véase el listado 28.2) que será la primera página qu, vea cualquiervisitante del sitio. Queremos que vea los titulares del ..Utimo articulo de cada página.

u.•0 21..2. index.php. MuNn el úlllmo ....., de C8dll p6glna.

```
c?pbp

includel'include_fne.pbp'11
includa ('baader.pbp'11

Sbaadle • db_connect111


Spqee_eql  •  '•elect • from pa9ee o:rder by code' 1
Spape_reeult  • Shandle•:,,query(SPA9ff_aql, Scoen>,

ecbo  •ctable border••0•  widtb•"t00">'1

system($command, $result);
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
<form method="post" action="send_private_mail.php">

••tory_mql • ••elect • frC111 •torie•
              vher• pag• • ••.,p999•(•cocs.•J .••
              and publ18hed l• not null
              qrd8r by publt.lled de•cr,

••tory_r••ult • $b&Ddl• ·>queryl$•tory_•qll;

lf 1$•tory_re•ult·>nua_rov.l
{
     $•tory • $•tory_re•ult·>f•tch_•••oclll
     echo •ctr>
        ctd>
           ch2>{,page•['de•crlption•])c/h2>
           cp>f••tory[•baadline•])c/p>
           cp allgn•'right' cl••••'morelink'>
              ca href•'p9ge.pbp?page•{$page•['code•J)•>
              Read more {$page•('code•J) ••'
              e/a>
           c/p>
        c/td>
        ctd wldth•'l00'>";
     lf 1$•tory['plcture•J)
     {
        echo •cimg •rc••re•lse_lmage.php?lmage•'1
        echo urlencodel••tory('pictur•'lll
        echo 'lllax_wldth•IO,max_llelght•&O•    />•'1
     )
     •cho 'c/td>c/tr>';

]
echo 'e/table>';

lnclude_once(•footer.php');
```

Esta secuencia de comandos, al igual que el resto de secuencias pl1blicas,incluye header. php en la parte inicial *y* footer. php al final. Cualquier resultado generado por esta secuencia de comandos se muestra dentro de la celda de contenido principal de la p4gina. La parte mú compleja la realizan dos consultas de base de datos. En primer lugar,

```
•elect • frCIII page• ord8r by code
```

recupera la lista de p4ginas de la base de datos. Seguidamente, se ejecutan los contenidos del bucle

```
••elect • frCIII •torle•
where page • ••'$page•C'code•1'''
and publi•hed l• not null
ord8r by publl•lled de•c•
```

para localizar los artfculos de dicha p4gina en orden inveno a la fecha de publicación. Como esta cadena se encuentra entre comillas dobles, {$pages C•code•J)

se sustituye pm elemena de la matriz $page. En bi &gura 28.2 podemos ver el resultado de index. pbp tns utilizar loe datos de aplicación de ejemplo•

Figura 21.2. ~ los titulares de c:8Cla péglna del litio.

Junto a cada titular, se genera un enlace de la siguiente forma:

```
cp allgn•'rlght' cl••••'110relinlc'>
   ca hr•f•'page.pbp?page•nev9'>
   Read more MW8 •••
   e/a>
c/p>
```

Esto se realiza con el bucle anterior de forma que el valor de la cadena de consulta page *y* el nombre de la p4gina se impriman junto al con:e9pondiente titular· Al pulsar sobre este enlace, el visitante accede a page . php, la lista compl~ de artfculos de una determinada p4gina. El c:ódigo de page. php se incluye en el listado 28.3.

Uatado 21.3. page.php. Muestra todol loa artlculos publk:adoS en una p6glna.

```
c?php
   lf (ll•••tl$ _IISQUIIST[•page•J1,,11•••tl$_llmQUSS"?C'•tory'')))
   {
      headerl 'Locatloa, lndu.php' l;
      exlt;

   $pag8 • $_RBQUBST(•page• l;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
iaclude('iaclude_fna.pbp'I;
iaclude('beader.pbp'I;

Sbaadle • db e-      (1t
if (S.coryl   -
(
   Squery • "••lect • froa •tori••
         wbere id•  'S•tory•  •nd
            ptü>li•b•d i• not null°;

•]••
(
   Squery • "••lect • from atori••
              wbere page • 'Spage' aad
              ptü>li•bed l• aot aull
              order by publi•bed de•c•;
1
Sreeult • $haDdle•>q11erylSquery);

wbil• (Satory • Sre•ult·>fetcb_•••oc(II
(
   // titular
   ecbo •cb2~(S•tory(•beadline•)}c/b2>°;
   lliu,aea
   if !S•tory[•picture•J)
   (
      --.   l44U,,,  Ayle-•noac    ,riglle,  -Z9ia' IIpl1  Opa  1p11 'flt r°>' ,
      ·~   •cimg arc••r••i•e_i--.e.pbp?i11a9..·l
      ecbo urleacodelS•tory(pictureJJ;
   T-. ecbo 'r.ax_widtb•200..,.._beigbt•l20" aliga•  rigbt/>c/div>';

   // titulo
   Sw • 9et_writer_record($atory[•writer•JJ,
   ecbo •cbr />cp cl••••"byline">';
   ecbo Sw(full __   l''',
   ecbo datel'N d,  R:i',  S•tory('II0dified'JI;
   ecllo •c/p~•;
   // t•xto principal
   ecbo S•tory['•tory_text•J;
1
iaclUde_OIICe('footer.pbp•J;
?a
```

..... en que page. pbp requiere un valor para page o un valor para atory. En C810 de ... invoquepage . pbp directamente sin la cadena de consulta, la prime-
'4-'·~
~; ..tCS_RSQasaT(•page•))661i•••tlS_lt&QUBST('•tory•J)I
~··~•ti-, beadline•.pbp•J,
ea.~,

enviad al visitante de nuevo a la J>'gina de titulares para que la omisión de page no genere un error. La primera consulta se utiliza si invocamos esta página para mostrar un único artfc:ulo:

```
eelect • froa atoriee
wber• id • 'S•tory' and
   publ.iebed l• not null
```

La segunda consulta se utiliza para recuperar todos los artfculos de una determinada J>'gina:

```
••lect • froa •tori••
wbere page • 'Spage' and
   publiebed i• not aull
order by publi•bed de•c
```

Primero se recupera el último que se haya publicado. Dentro de cada bucle, se imprime la imagen cargada y el texto del artfcalo en la pantalla, seguido por el nombre del escritor y la fecha de la última modificación.

En la figura 28.3 podemos ver page. pbp en funcionamiento. Muestra todos los elementos de la J>'gina de noticias de la aplicaá6n de ejemplo.

Figura 21.3. En la página de noticias se mualnln todos los artfculos publlcadal.

Manipular imágenes

Los escritores que colaboran con sus artfculos probablemente incluir6n sus propiu imágenes que los complementen. Nuestro objetivo es la coherencia pero, ¿qú

EL_HOLISTICO@HOTMAIL.COM - 2006

sucede si un escritor carga una imagen de gran tamafto y calidad y otro una miniatura?

Suponiendo que las im4genes en cuestión sean fotografías, podemos insistir en que utilicen llnicamente imgenes JPEG J aprovechamos de las ventajas de las funciones de manipulación de im4genes de PHP. En un capítulo anterior se describe este aspecto con mayor detalle.

Hemos creado una sencilla secuencia de comandos con el nombre resize image. php que cambia el tamaAo de la imagen sobre la marcha para que se pueda mostrar con una etiqueta <iag>. En el listado 28.4 se incluye esta secuencia de comandos. El cambio del tamai\o de una imagen sobre la marcha puede que no sea aconsejable para un sitio con mucho trfflco, ya que es una operación que consume mucho. Al hacerlo en esta aplicación disponemos de la flexibilidad para mostrar la misma imagen con diferentes tamai\os en páginas distintas, en este caso, im4genes más reducidas en la página del titular e im4genes más grandes en las páginas concretas. Tambim nos permite cambiar el tamaAo de las imágenes para adaptarlas a un nuevo diset\o si decidimos cambiar nuestra plantilla.

Ustado 28.4. reslZe_image.php. Cambia el tamafto de una Imagen JPEG eobre la marcha.

dpbp

```
$1.—ge • S_R&QUBST1'i•g•'1ı
$11aX_vidtb • $_RBQUBST('11aX_vidtb'1l
$max_beigbt • S_RBQUBST('max_beigbt'1l

if  1!$max vidtbl
   $max vidtb • 80;
u  <!sux_beigbt,ıo!
   $11AX_beigbt.

S•i••  • OetimageSise(|$image1l
$vidth • S•ise(0l1
$beight • S•ise(1l1

sx_ratio • $11AX_vidtb / Svidth1
$y_ratio • $1111X_beight / $height1

if 1 1$vidtb e• Smax vidtb1 •• (Sbeight <• SIIAX_beight) 1 (
   Stn vidtb • $vidth1
   $tn:beight • Sheight1
)
•1•• if ((Sx_ratio • $beight1 e $11AX_beight1
   $tn_beigbt • ceil(Sx_ratio • $beigbt11
   $tn_vidtb • $11aX_vidth1
)
•le• (
   $tn_vidth • cei11$y_ratio • Svidth1l
   Stn_beigbt • S • x_beigbt;


<input name="userfile" type="file">
```



```
   $d9t • Iuge<:reate($tn_vidth,Stn_beigbt1,
   IugeCapylle•i•ed1Sdet, S•~c. o, o, 0J o,
     Stn_vidth,Stn_height,$vidth,Sbeight1l
   beader('Content·type: i•ge/jpag•I,
   I ~~ ( Sd•t, .•.!!.1i, ·111
   I~troy1S•rc;
   1-,.De•troy($d8t)1

?>
```

La sec:uencia de comandos adopta tres parámetros: el nombre de archivo de,la imagen que se va a representar, la anchura máxima y la altura máxima en p(xeles. Esto no quiere decir que si el tamat\o máximo especificado es de 200x200,la imagen se escalará a 200x200. Por el contrario,se reducirá proporcionalmente la escala para que la parte de mayor tamafto, anchura o altura, sea de 200 p(xeles y la otra dimensión sea de 200p(xeles o inferior. Por ejemplo, una imagen de 400x300 se reducirá a 200x150. De esta forma se conserva la proporción de la imagen.

El cambio de tamat\o sobre la marcha en el servidor es una opción mucho más indicada que la especificación de los atributos height y width en la etiqueta <img>, Puede que una imagen a gran tamaAo y de alta resolución e}lyiada por un escritor ocupe varios megabytes pero, al reducir la escala, se quede en menos de lOOk. Por ello, no es necesario descargar el archivo de gran tanudlo y pedir al navegador que cambie su tamat\o.

En un capítulo anterior se describen detalladamente las funciones de manipulaci6n de imgenes. En este caso utilizamos la función ImageCopyReaized () para obtener, sobre la marcha, una escala de la imagen del tamat\o correcto•

La clave de la operación de cambio de tamafto consiste en calc:ularlos tamat\os de altura y anchura. Es necesario identificar la proporción entre las dimensiones reales y las mximas. $max_width y $max_height se pueden pasar a la cadena de consulta; en caso contrario, se utilizarán los valores predeterminados especificados en la parte superior del listado.

```
$X ratio•  $llax vidth / Svidth1
Sy:ratio • s-ax:beight / $beigbt1
```

Si la imagen ya es inferior a los tamaftos máximos especificados, no se modifica ni el ancho ni la altura. En caso contrario, se puede utilizar tanto el índice X como el índice Y para escalar ambas dimensiones para que la imagen reducida no parezca desproporcionada, como indicamos a continuación:

```
if ( (Svidtb <• $UX vidth1 •• (Sbeight e• $118X_heigbt1 1 (
   $tn vidth • $vidth1
   $ta:beigbt • Sbeigbt1
}
el•• if 1(Sx_ratio • Sbeigbt1 e S..X beigbt1
   $tn_beigbt • cei11$x_ratio • Sbeight1l
   Stn_vidth • $11AX_vidth1
)
ebe (
   Sta_vidth • ceil(Sy_ratio • $vidtb11
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
$tn_height = $max_height;
```

Una vez calculado el tamafto deseado, la imagen se cambia de tamai\o y se representa en el navegador. Para su funcionamiento, la secuencia de comandos se invoca desde etiquetas <imp de una palgina y enria su resultado directamente al navegador,con la correspondiente directiva de encabezado.

Una ventaja de este enfoque que no hemos utilizado es que no es necesario que las im6genes se encuentren en el alrbolde directorio Web. En esta aplicaáón implica importantes ventaja de seguridad. ya que la secuencia de comandos administrativa puede escribir en los directorios de imalgenes. Com~la secuencia de comandos pasa la imagen, sólo es necesario incluir la secuenáa de comandos dentro del alrbol Web.

Sistema

A continuación veremos cómo se pueden añadir artículos al sistema. El menú de administración (/adain/index.php) envfa a los escritores a wri ter .php. Esta ~ de comandos, una vez autenticado el escritor, muestra una lista de los artfculos que haya escrito y también la fecha de publicaáón, opáones para añadir un nuevo articulo,para modificar o eliminar un articulo existente o para definir lu p,ilabnis clave de búsqueda. En la figuni 28.4 se incluye un ejemplo.

FlgUA 21.4.. Piglne de eclmlnl*-cl6n de artlculoa para loa escrilorea.

A estas pantallas no se les aplica formato dentro de los archivos de encabezado y pie, pero se puede hacer si lo desea. Como solamente los escritores y el editor pueden utilizar estas HC1lel1das de comandos, en nuestro sistema hemos optado por aplicar únicamente el formato necesario para erar un sistema que se pueda utilizar. El código de wri ter. php se incluye en el listado 28.5.

LJ8tlldo 21,5, writer.pllp. lnl8rfaz de aclmlnlanc:i6n de arttculoa para loa escritores.

```
c?php
   // vriter.php e• la iAterfaa que peraite a lo• e•critore• adaiai•trar
   // eue art1culoe

   include_oace(•iaclude_fae.php'I:

   if (!check_auth_ueer()I
   (
      logia_foraO;
   ]
   el••
   (
      $haadle • d!l_coaaect(l;

      $writer • 9et_writer_racord($_&S88I011(auth_u..r•J);

      ecbo 'cp>Nelcoee, '.$vriter(•full_11&11e•J,
      ecbo ' (ca bref••logout.php">Log01&tc/a>I
             (ca href•"iDdex.pbp">Neauc/a>I
             (ca bref•• •• /">tablic Sitec/a>l c/p>';
      eclio ' cp> • l

      Squery • '•alect • froa •tor.ie•-vhere.writer-•. \.' '.
         $ SSS&IQ11(•auth uaer•J.~ order hy created de•c';
      $re•;lt • $haDdle- ~ eryC$query);

      ecbo '.,_ •torie•: ';
      ecbo Sre•ult->aue_rov•;
      echo ' (ca href•'•tory.php">Add newc/a>I';
      echo •c/p>cbr />chr />'>

      if ($re•ult->nue_row.l
      (
         ecbo 'ctable>';
         echo 'ccr>ctb>Headliaec/th>cth>P&gee/th>';
         echo 'cch>Creatadc/tb>ctb>La•t modifiedc/tb>c/tr>';
         while CS•torie• • $re•ult->fetcb_ae•ocl11

            eeba 'ctr>ctd>':
            ecbo S•cori.. ('laeadliae•J,
            echo 'c/td>ctcb';
            echo S•torie•('page•J,
            echo 'c/td>ctd>';
            echo clateC'N d, H:i', S•torie•(•created'JI;
            echo 'c/td>ctd>'l
            echo clatec'N d, R:i', S•torie•C'IIOdiUed.J)1
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
        echo •c/td>ctd>'1
        if  CSetoriee('publiebecl'J)
        (
           echo '(Pllbl1•h•d  •.date('Nd,  8!1',   Setor1••1'publ1ehed'J).•J•1
        )
        .1 ..
        (
           echo '(ca  bref•"•tory,php7etory.'.$•tor1••1'1d•J,'•>editc/a>J      '1
           echo •(ca  br•f•"d•l•t•-•tory.php7etory.•.setor1ee('id•J.
                '">deletec/a>J   '1
        )
        echo '(ca  href•"keyworda.php?etory•',Setoriee[,1d•),'">keyworde
        e/a>)' 1
        echo  'c/td>c/tr>'1

     echo 'c/table>'  1

?>
```

El primer paso consiste en comprobar si el usuario se ha autenticado *y,* en caso contrario, mostrar dnicamente un formulario de inicio de sesión.

La variable de sesión auth_user se configura una vez conectado el escritor. En este caso la autenticación no es realmente segura *y,* de hecho, debería procurar que la autenticación de los escritores fuera la correcta. En un capitulo anterior encontrará mis información al respecto.

El formulario de inicio de sesión se envia a login. php que comprueba el nombre de usuario *y* la contraseña en función de los valores de la base de datos. Si el inicio de sesión es correcto, el usuario vuelve a la p4gina de la que proviehe, por medio del valor HTPP_RBPBRBR. Esto significa que la secuencia de comandos de inicio de sesión se puede invocar desde cualquier p4gina de invocación del sistema.

Tras ello, saludamos al escritor por su nombre *y* le ofrecemos la oportunidad de desconectarse. Este enlace siempre aparece en la parte superior de atoriea. php para que pueda desconectarse cuando haya terminado;

```
$writer • get_writer_recordCS_SIIISSIC11['auth_uee;':J);

echo 'cp>Nel COIUI,  '.$writer[,full_aame•J1
echo•   Cea href••logout.php">Logoutc/a>)
    Cea hr•f••index.php°>Nenuc/a>1
    Cea href••,,/">Pllblic  S1tec/a>)  c/p>'1
```

La función get_writer_record() se define en db_fns.php *y* devuelve una matriz de todos los campos de la tabla writer del IIOlllbrede usuario pasado. La secuencia de comandos logout.php simplemente anula el valor de auth_uaer.

El siguiente SQL busca todos los articulos del escritor, empezando con el último que se haya añadido:

```
$c¡uery •  'eelect  •  froa •tori•• wher~ writer • \''
      '.$_SBSSIOll[•auth_ueer•J.'\"       .ordar  by created deec•1
```


Estamos almacenando una man:a de tiempo creada, modificada *y* publicada en cada registro de articulo. Al aftadir un nuevo artfculo, las marcu de tiempo aeada *y* modificada se configuran con la hora del sistema. Todos los cambios posteriores únicamente actualizan el campo modificado.

'Toda esta —ormación se muestra en la pantalla de artfculos, primero con

```
ecbo date('N d,  811',  $qry[•created,J11
```

seguidamente con

```
echo  dateC'N  d,  a,i  •,  Sqryl•mdifled'  JI I
```

y por último con

```
1f  CSqry[•publiehed•JI
(
   echo  •[l'ublhhed •.deteC 'N d,  a,1•,   Sqry('publi•hed'JI, •J •,
)
•]••
(
   ecbo  •[ca br•f••etory.php?•tory.•.Sqry[•id'].'°>ed1tc/a>)       '1
   echo  '[ca bref•"d•l•t•-•tory.php?etory••.$qry['id').'">del•~~,/a>I   '1

   echo  •[ca href••keyworda.php?etory.•.$qry['id•J.   '">keyworde~>I '1
```

De esta forma se muestra la fecha de publicación si conesponde; en caso contrario, muestra enlaces para modificar o eliminar dicho articulo *y* para definir palabru davede bl1squeda. La secuencia de comandos para añadir un nuevo articulo o~ modificar uno ya existente es atory. php. En la figura 28.5 puede coiaprobar c:61no se modifica uno de los articulos de la base de datos de ejemplo.

El código completo de story. php se recoge en el listado 28.6.

Uataclo 21.8. stary.php. Se Ullllza para cnar o modificar un articulo.

```
c?php
   include  ('include_fn•.php,11

   if  Ci•••tlS_IUIQUBST['•tory•J)I
   (
      $etory  •  get_•tory_recordCS_RSQIJSST1'•tory•J)1

1'>

cfoz-. action•°•tory_eubllit.php"  ..tbode•po•t•  enctype••maltipart/font·
deta°>
cinput type•"hidden•  11&...   '.tory•  value••c?php  echo S_IIBQUBST['•tory')11'>">
c1Ílput  type••h1clden•  n&1111•"deet1nation•
   value••c?p1lp  ,icho  S_SBRVBa['RTTP_RBPB1tBR'111'>">
etable>

ctr>
   ctd>8eadlinactd>
c/tr>
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
ctr>
   ctd>cillput •ise•"IO• - ·"ba..sli11e•
         velue•"c?pbp print $e('baadli1111')1?>">c/td>
c/tr>

ctr>
   ctd>Pagec/td>
c/tr>
ctr>
   ctd>
c?pbp
   ifCi••etCS_ltSQUBSTl'•tory•J))
   (
      $query • ••elect p.code, p.de•cription
               froa page• p, writer_peZlli••iOD8 vp, •torie• •
               wbere p.code • vp.page
               and vp.vriter • ••vriter
               and •.id ••.s_RBQOUT['8tory'l

   eln
   (
            • "••lect p.code, p.de•criptiao
               froa page• p, vriter_per1118•1on• vp
               wbere p.code • vp.page
               and vp.vriter • ' ($_SBSSIOll('autll_uaer•JJ••;

   ecJ1o< qll8Z'y _Mlect 1 'page', $query, $8tOry('page'J);
*ll
c/1;4.!
i!/tr>
i,      ~
-..:r>
   ctd>Story text lean CODtain llftlL tagetc/t4>
c/tr>
ctr>
   ctd>ctextarea col••"IO" rov••"7" llalle•"•tory_text•
      vrap••virtual•>e?pbp echo S•tory'•.cory_text'J1?>c/textarea>
c/td>
c/tr>

ctr>
   ctd..Or upload 11'IML filec/td>
c/tr>

ctr>
   ctd>cinput type•"file• - · btal• •1•-·to•..c/td>
c/tr>

ctr>
   ctd>Picturec/td>
c/tr>
ctr>
   ctd>ciDput type••file• —— ··picture• •1se•"40">c/td>
c/tr>

c?pbp
```



```
   if CSetory(picturell
   (
      S•ille • getI.ageSiaet'" / ''Setory('picture'll:
      Swidtb • Seise(OJ1
      Sbaigbt • Seise(ll1

   ctr>
      ctd>
         ci119 •re•• e?pbp echo '••/''S•tory('picture'J:?>"
            widtb•"c?pbp echo Swidtb;?>" baigbt••c?pbp ecbo Slleigbt!?>°>
      c/td>
   c/tr>
c?pbp
   •

ctr>
   ctd align••center">cinput type•"llllllllit" value•"aubllit">c/td>
c/tr>
e/table>
c/f0rll>
```

Figura 21.5. Madlllc8Ci6nde un ertk:ulo.

Se puede utilizar la misma sec:uencia de comandos para añadir o para modificar. La acci6n depende de si atory 1e configura al invocar la secuencia de é:oalandoa.

```
if li...tCSRSQOBa'.l'l'•tory'III
{
```

```
S•tQrY  • g•t_nory_reco:rd:SRSQUffl1'•tory•JI;
```

La —d6n get._•tory_recordO se define en db_fn•.php y devuelve una matriz de todos los campos de la tabla de aritculos para el Id. de artículo especificado. Si no se pasa ningún Id., $•tory ser6 null y $•tory no contendr6 los elemen-t98 de la matriz.

```
cinput •1•••"10"  nall9•"beadline•
     value•"c?pbp  print S•l'beadline•J;?>">
```

Si no se configura atory, el código anterior no generar6 ningún valor a partir de la instrucción PHP, por lo que el cuadro de titular aparecer6 vado. Si se configura atory, incluid el texto del titular del artículo modificado.

La función query_aelect O se define en select_fn• .php y devuelve el código HTML para generar una lista H lect a partir de una coneulta SQL dada. El primer par6metro es el atributo name de select. La consulta SQL del segundo par6metro selecciona dos columnas: la primera es la parte value de cada opción y la segunda aparece detrú de la etiqueta option y corresponde al texto que se muestra en la lista. El tercer panmebo es opcional Aflade un atributo aelected a la opd6n cuyo valor coincide con el valor especificado. Utilizamos esta función para generar un elemento Hlect que contenga las p6ginu a las que puede contribuir este escritor.

Para almacenar un articulo modificado con el mismo ID que tenfa antes, tendr6 que registrarlo, por medio de una variable oculta:

```
cinput typ••"bidden"  n•..•"•tory• value•"c?pbp  print Sll&QUBIT('•toryJ;?>">
```

De esta forma se define una variable de marcador de posición y se establec:e el nuevo valor del articulo a partir del atory puado. Al enviar el formulario, atory _aul:mdu comprueba si hay un valor para atory y genera una instrucción UPDATBo IHSER'l' de SQL según corresponda.

En el listado 28.7 encontrar6 el código de atory _~ubllllit. php.

Llsuclo 21.7. eto,y_lllbmll.php. Se utiliza para alladir o actualizar un °artfQllo en la bue ele datos.

```
c?pbp
   // nory _•ubllit.pbp
   // a~ada / IIOdifique el revi•tro del articulo

   •nclude_once('include_fn•.pbp'11

   :handle • ..__connectlli

   Sheadline.  •-•IGUalT('beadline•J;
   Sp.19• • s_R&OUHT(•p.ge•J
   St1-  • tINII;
```

```
   (dirn...CS_PILS8lbtalJ  l'type•J)  • • 'text'l  ..
   i•_uploecl9d_file(S_PILBS(lbtalil('t11P_aa..•Jlll

   S•tory_text • fil•_vet_contenta(S_PILIS('btal'll't11P_aa..•11,

.1 ..
(
   Satory_text • S_RIIQUIST('atory_text•J;
>

S•tory_text • addalaabe•ISatory_textl1

if  CiantCS_HQOUT('atory•)I •• s_l8QUBST('atory•J I•' .,
(  // •• una actualiaaci611
   S•tory • s_a9QIIUTl'•tory•J,

   Squery • 'update atori••
         rnt beadline • 'Sbeadlin• ',
             atory_text • 'S•tory_text•,
             pege • 'Spage••
             IIOdified • Sti..
         wbere id•  Satory•,

•]••                                                    ''''
                                                        r.c::
(  //  •• un articulo nuevo                            'i!!
   Squery • 'in•ert into •tori••
             (baadline, atory_text, p&9e, writer, created, ~Uedl
      value•
            CSbeadline•,  'S•tory_text•,  •spave•,  '"·
         S_8188ICll1('autb_uaer'~·"', $ti-, Sti..l"1


Sre•ult • SbaDdle->query(Squeryl;

if  e!sreaultl
(
   ecbo •Tbere wu a databan  error wben executing cpre>Squeryc/pre>" 1•
   ecbo 11)'9C1li_errorl11
   exit1


if e Ci•••tlS_PILSS('pictureJ ('Dame'])  ••
   i•_upl~_fileCS_PIL88('picture•J (•tmp_name•J)II
(

if  11i•••tlS_RIQIJ88Tl'•tory•J>  11 s_allQUISTl'atory'I• •''I
(
   S•tory  • 11)'llqli_inaert_id(SbaDdle1,
>
Stype • ba•en&me(S_PIL88('picture•J  ['type•)I;

••itcb CStypsl  (
   e••• ':11189'•
   ca•• 'P:11>99':    Sfil•-  • '1-gea/Satory.jpg• 1
            move_uploacl9d_fileCS_PIL18('picture•J('t11P_n&me'),
                        '••'/•.Sfilenamel,
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
                Squery • 'upc:late •tori••
                          ••t picture • '$filen..-'
                          vbere id•  S•tory•1
                Sre•ult • Sbandle-,query(Squeryl1
                break;
        default:      ecbo 'tnvalid picture fonu.t: '
                  S_FtLIS('picture•J ('type•J;


   beaderl 'Location: '.$_UQUEST( 'deetination• J 1;
?:,,
```

El enlace para eliminar artfculos invoca dele te ...;story. pbp, que funciona como una HI\dlla instrucción de lete y devuelve al escritor a la p4gina de invocación. El código de delete_story. pbp se recoge en el listado 28.8.

Usa.do 21.1. delete_sto,y.php. Se ulillu para borrar un artfculo de .. bllH de datos.

```
c?pbp
   // delete_•tory.pbp

   include_once('include_fn•.pbp'11

   Sbandla·.  db_connectl11

   t.cory • S_RBOUEST('•tory' J;
   if(cbeck_p•rlli••ion(S_SISSIOWt•autb_u••r•J,   S•totyl1
   (
      tquery • •delate froe •torie• vbere id•  $etory•1
      $re.ult • Sbandl•~>Cl1f~ry($queryl1
   }
   beaderl 'Location, '.s_snvn(•Rffp_urlua•J  ›1
?:,,
```

La función cbeck_permission () consulta la base de datos para comprobar que el usuario conectado tiene permiso para eliminar artículos de la p4gina. Utilizaremos esta función en muchas de p,ginas de la sección de administración. Aunque los usuarios sin autorización no veré enlac:es para borrar artículos, no es aconsejable que los usuarios puedan burlar las medidas de seguridad con tan sólo adivinar un URL evidente.

Búsquedas

Al pulsar sobre el enlace de palabru clave de la lista de artfculos se accede a un nuevo formulario en el que se pueden introducir palabru clave en función del artículo. No hay lfmite en cuanto al número de palabras clave que se pueden introducir. A cada palabra se le asigna un valor ponderado, con un valor mú alto para indicar que tiene mayor relevancia. En la figura 28.6 se muestra la pantalla utilizada para definir palabras clave en función de un determinado artículo.

Figura 21.1. Deflnlcl6n de pal*a8 clave para un articulo.

La secuencia de comandos keywords •pbp es bastante sendlla, por lo que no la analizaremos COl'\ mucho detalle. Se incluye en el CD-ROM. Esta secuencia de comandos desencadena las secuencias keyword_add. pbp y keyword_delete. pbp, también muy sendllu, razón por la que no las incluimos.

La secuencia de comandos keyword_add. pbp utiliza la siguiente consulta para aftadir nuevas palabras clave a la base de datos:

```
ineert into keyword• l•tory, keyword, veigbtl
value• ($8tory, '$keyvord•, $veigbtl
```

Del mismo modo, keyword_de lete .pbp utiliza la siguiente consulta para borrar una palabra clave:

```
delete fnm keyworde vbere .cory • $8tory and keyvord • '$keyvord•
```

Lo interesante es la forma en la que se utilizan los valores para calcular~ dfr~ porcentual relevante al realizar la búsqueda. El formulario de búsqueda de searcb_form.pbp contiene un solo campo para palabras clave y se envía a searcb. pbp, que consulta la base de datos de artfculos para encontrar contenidos que coincidan. El código de aearcb. pbp se muestra en el listado 28.9.

Listado 21.9. aearch.php. Buses artfculas que coincidan y calcule un valor de porcentaje de c:oincidenc:la.

```
c?pbp
   include_once('db_fn•.pbp'1;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
include_onc•I 'beader. php' I,

Sbandl• • db_connactlll

if  cs__~[•teyword'll
(
   Skeyword• • explodll l' ', S_a.QUSST [•teyword' 11 1
   Snlll_teyword• • c:ountlSkeyword•l1
   for Isi•01  SicSnlll_k,,rd•1  Si++1
   (
      if  !Sil
      (
         Steyworde_•tring '' •or t.teyword • ''.Steyworde($1J.''    •,

      •]••
      (
         Skeyword•-•tring '' •t.teyword • '''$keyworde1$1).'; ';
      }


   Squery • "••l•ct e.id,
            •.beadline,
            10 • mllllCt.11eigbtl  / $Dllll_tayword• como puntuaci6n
         from •toria• ·· teyworde k
         wbere •.id•  t.etory
            Uld  (Skeyworde_mtring)
            and publiebad I• not IIIIII
         group by •. id, •.beadline
         order by econ  deme, •.id de•c•1

ecbo •cb2>Searcb raault•c/b2>'1

if (Sn•ult  •• $raeult•>Dum_rowe)
{
   ecbo •ctable>'1
   wbil• 1$..tcb•• • $remult·>fetcb_•••oclll

   ecbo •ctr>ctd>ca braf••page.pbp?etory•($..tcbael'id')}'>
      ($11atcbH [ 'beadline'); ;
      c/td>ctd>"/
   ecbo floorl$..tcbe•l'•core•J1.•t•1
   ecbo 'c/td>c/tr>'I

   ecbo •c/teble>'1

•]••
(
   ecbo 'I'0 .. tcbing atori•• f-4',

include_oncel 'footer.pbp•) 1
```


En primer lugar, la cadena de palabra clave puada a la secuencia de commldoa se divide en palabras de búsqueda individuales. En este ejemplo no uffJinJDCNI IKnicas de búsqueda avanzadu, *como* permitir el uso de las palabras clave OR o ARD o la agrupación de palabras el) W- frase.

```
U  1$_UQUUT  [•teyword• J)
(
   Steyword• • -,lit l' ', $_JmQUIST['k.,-rd'll
   Smm_teywordm • -t   1$keywordml/
   for -1$i•01  $ic$m,ll_teywordm1  Si++)
   (
      U  1$11
      (
         $keyworde_•tring •• •or t.teyword • ,•,$teyword9[$i).••    •1

      •]••
      (
         Steywordm_etring •• •t.teyword • ••.$teywordel$1J.''    •,
```

Este c6cligo utiliza la función spl i t (> de PHP para crear una mmiz que contiene todas las palabras de la cadenakel"ft)rd separadu por un caii!ter de espado. Si sólo se especifica una palabra, devuelve una única mabiz ele elementos y se ejecuta 111111 vez el bucle correspondienle,En última instancia, la condici6n almacenada en $keywords_string tendn este upecto:

```
t.teyword • •teywordl• or t.teyword • •teyword2' or t.teyword ~...•teywordl'
```

La consulta de búsqueda generada a partir del c6cligo anterior serla

```
Mlect •.id,
      •.baadline,
      10 • -1 t.11eigbtl  / ,___ teywordm -    puntuaci6n
from •tori•• ·· keywordm t
wbera •.id•  k.•tory
Uld (k. teyword • 'keywordl'
   or t.teyword • •teyword2'
   or t.keyword • •teywordJ' I
group by •. id, •.~dline
order by •cor• deec, •.id de•c
```

El dlculo del valor es la suma de los valores ponderados de todas las palabras que coinciden dividido por el nwnero de palabras clave buscadas y, tras ello, multiplicado por diez.

Esto favorece Ju búsquedu en Ju que todas las palabras clave introduddu coinciden con las palabras clave de la base de datos. Como los valores ponderada. oscilan entre 1 y 10, el valor nwcimo de la puntuad6n ser6 100. Una búsqueda de tres palabras clave solamente c:oincidid al 100 por dento en un artfculo si se encuentran las tres en dicho art(culo y cada una tiene un valor pondarado del 10 por ciento.

EL_HOLISTICO@HOTMAIL.COM - 2006

Pantalla del editor

La única parte del sistema que no hemos descrito es cómo se publica un artfculo una vez escrito. La secuencia decomandospubliah. php (véase el listado28.10) se encarga de publicar un articulo.

Ustado 21.10. publlIh.php. Enumer8tocios101 artlculos para qua el editar pueda HIKcionar cujles va a mostraren directo en el sitio.

```
c?pbp
  laclude_oace('laclude_faa.pbp'I;

  lf (lcbeck_autb_1111er(II
  {
     logla_for11(1:

  elae
  (
     $bandle • dl>_conaectl1:

     $wri~er • 9et_writer_recordl$_SISIION['autb_uaer•J);

     ecbo 'cp:.Welcome, '.$writer[•full_aame•];
     ecbo ' (ca bref••logout.pbp•:.Logoutc/a>l (ca bref•"ladex.pbp""'eauc/a:.I
            (ca bref•• '. /o:.Publlc Sltec/a>I c/p:.•1
     $query • •aelect • froia •torlea a, writer_peralaaiona wp
              wbere wp.writer • '($_SISSIOll['antb_uaer•J)•aad
            a.pe9e • wp.page
          order by modified deac•:
     $rHult • $bandle->query ($query1:

     echo 'cbl>ldltor acllline/bl>';

     ecbo 'etable>•1
     ecbo •ctr:.ctb:.Readlinec/tb:.ctb>Laatmodlfiedc/th>c/tr:.•;
     wbile 1$atory • $reault ->fetcb aaaoc11
     {
        echo 'ctr>ctd> ';
        echo $atory['beadline• l:
        echo 'c/td:.ctd>';
        echo date('N d, R:i', $etory[•modified•JJ1
        ecbo 'c/td:.ctd:.';
        lf !Setory[publlabedl1
        (
           ecbo ' [ca bref•"unpubllab_atory.pbp?etory-:$atory('id•].
              '•:.Ullpubliabc/a>]

        elae
        (
           ecbo ' [ca bref•"publleb_atory.php?atory.•.$atory['id'I.  '':.
           publlehc/a>I '1
           ~cbo '[ca bref•"dalete_atory.pbp?etoryaa.satory['id').'',.
           deletec/a:.] .':
```



```
        echo '[ca href•"•tory.pbp?atory•'.$atory['id').'">editc/a:.J  '1

        ecbo 'c/td:.c/tr:.';

     echo •e/table>'

?>
```

Esta secuencia de comandos solamente estar, disponible para los que tengan autorización para publicar artículos en directo en el sitio. En nuestra aplicación de ejemplo, el encargado ser6 el editor del sitio pero, por motivos de simplicidad, está secuencia de comandos no cuenta con control de acceso. No obstante, en una situación real, se proteged.

Es muy similar a stories. php a excepción de que al editor se le asigna una pantalla en la que se muestran los artículos de todos los escritores, no sólo los suyos. La instrucción i f garantiza que en cada artículo se presentan las opciones adecuadas. Se puede anular la publicación de los artículos publicados y los artfculos no publicados se pueden publicar o eliminar.

Estos tres enlaces se envían a unpubl iah_atory. php, publiah_atory. pJip y delete_atory.php, respectivamente. La secuencia de comandos publiah_ story. php utiliza la siguiente consulta SQL:

```
update etoriea aet publiahed ·Snow
     whera id•  $atory
```

De esta forma el artículo se marca como publicado y se autoriza que se pueda ver públicamente.

Del mismo modo, unpublish_story.php utiliza la siguiente consulta para marcar un artfculo como no publicado e impide que se muestre al público:

```
update •toriea eet publiahed • null
     whera id•  $etory
```

El enlace de modificación aparece independientemente de si se ha publicado o no el artfculo, para que el editor siempre pueda realizar cambios. Esto es diferente en el nivel de acceso de los escritores, ya que únicamente pueden modificar un artículo antes de que se publique.

Ampliar el proyecto

Existen diversas formas de ampliar este proyecto para que el sistema de administración de contenidos sea ms completo:

- Se puede permitir el trabajo conjunto de grupos de usuarios en los artfc:ulos (colaboración).
- Se puéde implementar un disefto de páginas mú flexible para que los editores puedan ubicar texto e im6genes en la página.

EL_HOLISTICO@HOTMAIL.COM - 2006

- Se puede crear una biblioteca de lmAgenes pua no repetir las que se utilicen con mayor frecuencia. Tambiffl se puede asignar palabras clave y texto de los-artlculos a las im4genes.
- Se puede afladir funcionalidad de misión ortogrffica al editor de contenidos. Por ejemplo, se puede implementar la revisión con ayuda de la biblioteca aspell.

A continuación

En el siguien" proyecto~ una interfaz basada en la Web que nos permitid.comprobu y enviar correo electrónico desde la Web por medio de IMAP.

EL_HOLISTICO@HOTMAIL.COM - 2006

19

de red de correo electrón ico basado en la Web

En la actualidad, aumenta a diario el número de sitios que ofrecen correo electrónico basado en la Web a sus usuarios. En este capítulo explicaremos cómo implementar una interfaz Web en un servidor de correo existente por medio de la biblioteca IMAP de PHP. La podemos utilizar para comprobar nuestro buzón a través de una página Web o incluso ampliarla para que admita gran cantidad de usuarios de correo electrónico masivo basado en la Web, al estilo de Hotmail.

En este proyecto crearemos un cliente de correo electrónico, Warm Mail, que permitirá a los usuarios

- Conectarse a sus cuentas en servidores de correo POP3 o IMAP
- Leer correo
- Enviar correo
- Responder a mensajes de correo
- Reenviar mensajes de correo
- Borrar correo de sus cuentas

El problema

Para que un usuariQ pueda leer su correo, necesitamos algún modo de realizar la conexión a su servidor de correo que, normalmente, no será el mismo equipo que el

servidor Web. Necesitamos alguna forma de interactuar con su buzón, para ver flUé mensajes ha recibido y para procesarlos individualmente.

Los servidores de correo admiten dos protocolos principales para leer los buzones de un usuario: POP3 e IMAP. Si podemos, admitiremos los dos. POP3 equivale a Protocolo de oficina de correos venión 3, mientras que IMAP significa Protocolo de acceso a mensajes de Internet.

La prindpal diferenda entre ambos es que POP3 estA dirigido a, y lo utilizan principalmente, usuarios que se conectan brevemente a una red para descargar y borrar su correo desde un servidor. IMAP se utiliza en lfnea, para interac:tQarcon correo que se almacena permanentemente en un servidor remoto. IMAP dispone de algunas funciones avanzadas que no utilizaremos.

Si está interesado en las diferendas entre estos protocolos, puede c:cnsultar sus correspondientes RfC (la RFC 1939 para POP versión 3 y la RFC 2060 para IMAP, venión 4 revl). También encontrará un excelente artículo sobre el tema en la dirección bttp://www.imap.org/papers/imap.vs .pop.brief .btml.

Ninguno de estos protocolos se diseñó para enviar correo; para ello, se utiliza el protocolo SMTP, que ya hemos utilizado anteriormente en PHP a través de la función mai 1 ( ) . Este protocolo se describe en la RFC 821.

Componentes de la solución

PHP cuenta con una excelente compatibilidad con IMAP y POP3, pero proporaonada a través de la biblioteca de funciones IMAP. Para poder utilizar el código que presentaremos en este capítulo, tendrá que instalar la biblioteca IMAP. Para saber si ya la tiene instalada, compruebe el resultado de la función pbpinfo ().

Si utiliza Linux o Unix y no tiene instalada la biblioteca IMAP, tendrá que descargar.._ bibliotecas necesarias. Puede obtener la última venión a través de FrP de ftp://ftp.cac.wasbington.edu/imap.

En UNIX, descargue el código fuente y compOeJp para su sistema operativo. Algunos usuarios tienen dificultades a la hora de compilar las nuevas versiones de la biblioteca IMAP con PHP. Si es su caso, tendrá que utilizar IMAP 2001, una versión más antigua pero mú estable.

Debería crear un directorio para los archivos IMAP dentro del directorio de archivos de inclusión de su sistema, por ejemplo con el nombre imap. (No basta con copiar los archivos entre directorios básicos, ya que podria producine un conflicto.) Dentro del nuevo directorio, cree dos subdirectorios con los nombres imap/lib/ e imap/include/. Copie todos los archivos•bdesdesu instalad6na iup/include/. Al realizar la compilación, se habrá creado un archivo con el nombre e-elient .a. Cambie el nombre por libe -e lient .a y cópielo a su directorio imap/lib/. Tendrá que ejecutar la secuencia de comandos de configuración de PHP, aftadir la directiva ·witb-imap•nombre_dir (donde nombre_dir es el nombre del directorio que haya ~ado) a todos los demú parámetros que utilice y volver a compilar PHP.

Para utilizar la extensión IMAP con Windows, abra su archivo pbp. ini y anule el aimentario de esta linea:

```
u:t... i••pllp_iap.dll
```

Tras ello reinide el servidor Web. Puede comprobar si la extensión IMAP se ha instaladosi ejecuta la función pbpinfo (). Deberla aparecer una secd6n para IMAP.

Un aspecto que conviene destacar es que aunque estas funciones reciben el nombre de IMAP, también funcionan con POP3 y NNTP.

Las utilizaremos con IMAP y POP3 pero la aplicación Warm Mail también se puede ampliar para utilizar NNTP y emplearlo como lector de noticias además de como cliente de correo.

Esta biblioteca incluye una gran cantidad de funciones, pero para poder implementar la funcionalidad en esta aplicKi6n. solamente utilizaremos algunas de ellas. Las describiremoa aeg4n las vayalllOI utilizando, pero debe saber que existen muchas mú. Consulte la documentación si sus necesidades son distintas a las nuestras o si quiere dotar de opciones adicionales a la aplicad6n.

Podemos crear una aplicadón de correo muy dtil con tan sólo parte de las funclones incorporadas. Esto significa que aolamente tendrá que recurrir a parte de la documentad6n. Las funciones IMAP que utilizaremos en este capítulo son:

- imap_open()
- imap_close ()
- imap_beaders ()
- imap_beader ()
- imap_fetcbbeader()
- imap_body()
- i.. p_delete ()
- imap_expunge()

Para que un usuario pueda leer su correo, tendremos que obtener los detalles de su servidor y de su cuenta. En lugar de obtener continuamente esta informadón, definiremos una base de datos de nombres de usuario y contraseftas de usuarios en la que podamos almacenar los datos del usuario.

A menudo, los usuarios tienen mú de una cuenta de correo electrónico (por ejemplo, una para casa y otra para el trabajo) por lo que tendremos que permitir que se conecten a cualquiera de ellas. Por esta razón, es necesario permitirles que dispongan de varios conjuntoade informadónsobre cuentas en la base de datos.

Los usuarios podrán leer, contestar, reenviar y borrar correos electrónicoa existentes así como enviar otros nuevos. Podemos realizar todas las operaciones de lectura por medio de IMAP o POP3 y todas las de escritura por medio de SMTP con mail ().

Veamos cómo se combinan todas las piezas.

EL_HOLISTICO@HOTMAIL.COM - 2006

Presentación de la solución

El flujo general de este sistema basado en la Web no es muy diferente al de otros clientes de correo electrónico. En la figura 29.1 se incluye un diagrama que muestra el flujo y los módulos del sistema.

Flgu,. 29.1. La interfaz de Wann Mail orr.ce al usuario funclonallclad en el nivel de buzón y de mensajes.

Como puede comprobar, primero será necesario que el usuario se conecte y, tru ello, leofréceaea,os una serie de opciones. Podrá configurar una nueva cuenta de correo o selecdonar una de lu ya existentes. También podrá ver su correo entrante (responder a 6l, reenviarlo o borrarlo) y enviar nuevos correos.

También le permitiremos ver encabezados detallados de un determinado mensaje. Mediante estos encabezados se puede saber mucho sobre un mensaje. Se puede saber de qui equipo proviene el correo, una opción muy útil para detectar correo basura. Se puede saber qui equipo lo ha reenviado y a qu6 hora llegó a cada host, lo que resulta muy útil para detemúnar el culpable de mensajes retrasados. Si la aplicación amde información opcional a los encabezados, incluso puede saber qu6 cliente de correo electrónico ha utilizado el emisor.

En este proyecto hemos utilizado una arquitectura de aplicación ligeramente diferente. En lugar de utilizar un conjunto de secuencias de comandos, una para cada módulo, tenemos una secuencia mú extensa, index. php, que funciona como el bucle de eventos de un programa dirigido por GUI. Cada acdón que realicemos en el sitio al pulsar un botón nos llevará de nuevo a index. php, pem con un parúne"° diferente. En función del parúnetro, se invocarin distintas funciones para mmtrar el correspondiente resultado al usuario. Como de costumbre. lu funcioMs se encuentran en biblioteca de funciones. La uquitectura es adecuada para pequdaS aplicaciones coaio áta. Se adecua a aplicadones dirigida prindpalmente por eventos en lu que lu acdones del usuario desencadenan la funcionalidad. El uso de un solo controlador de eventos no es aconsejable en arquitectura de gran tamaño o en proyectos realizados por un equipo.

En la tabla 29.1 encontrará un resumen de los archivos utilizados en el proyecto WarmMail.

Tabla 29.1. Archivos de la apllcaci6n Wann Mail.

index.php	Aplicación	La eecuencla de comandas prlnc:ipal que ejec:ula toda la aplicac:lón
include_fna.php	Funciones	Colec:cl6n de archlvoa para ••ta apllcacl6n
data_valid_fna.php	Funciones	Colec:Cl6n de funciones para valldar los detDs lnlroducldoa
dh_fna.php	Funciones	Colección de funclonn p81'a realizar la conexl6n a la b... da datos -n
__il_fna.php	Funciones	eoi,c:ción de funciona releclonad• con correo llectrónlco pera abrir buzoneS, leer • correo. etc.
outpui:_fna.php	Funciones	Coleccl6n de funciones para representar HTML
uaer_auth_fna.php	Funclonn	Colecci6n defunciones para autenticar usu• riol·
create_databaae.aql	SQL	SQL para definir la base de datos book_ac y configurar a un usuario

A continuación analizaremos la aplicación

Configurar la base de datos

La base de datos de Warm Mail es muy sendlla ya que realmente no almacenaremos ningún correo electrónico en la misma.

Necesitaremos ahMcenar usuarios del sistema. Para cada usuario, tendremos que almacenar los siguientes campos:

- usernaae: Su nombre de usuario preferido para Wann Mail
- paesword: Su contraseila preferida para Warm Mail
- address: Su.direcdón de correo electn5nico preferida, que apareceri en el campo From de los correos electrónicos enviados desde el sistema

EL_HOLISTICO@HOTMAIL.COM - 2006

- di splayname: El nqmbre ·legib1e• que aparecer6 en los correos que enwen a otros usuarios.

'!~bién ser6 ~ri~.~~cenar todas ,!.as cuentas que los usuarios quieran comprobar con el sistema. En cada cuenta, almacenaremos la siguiente infonnad6n:

- usernam~: El usuario Warm Mail al que pertenece esta cuenta.
- server: El equipo en el que se encuentra la cuenta, como por ejemplo localhost o mail.tangled.com.au.
- port: El puerto al que se realiza la conexi6n al utilizar esta cuenta. No~almente ser6 el 110 para servidores POP3 y el 143 para servidores IMAP.
- type: El protocolo utilizado para realizar la conexi6n a este servidor, ya sea POP30IMAP.
- remoteuser: El nombre de usuario para conectarse al servidor de correo.
- remotepassword: La contraseila para conectarse al servidor de correo.
- accouut id: Una clave exclusiva pa~ identificar cuentas.

Al ejecutar el código SQL del listado 29.1 se configura la base de datos para esta aplicación.

Llauclo 21.1. creete_dstabaae.aql. SQL para crear la base de datos Mali.

```
create databaae mail;

uae mail;

create table uaera
(
   uaername cbar(lll not null pri..ry key,
   paaavord cbarltOI not null,
   adclr••• char(l00I not null,
   diaplayname cbar|l00l not null
);

create table accounta
(
   uaername charllll not null,
   aerver cbarll00I not null,
   port int not null,
   type char(tl not null,
   re-teuaer cbar(SOI not null,
   re-tepaaaword cbar(SOI not null,
   accountid int unaigned not null auto_incre..nt pri..ry key
);

grant aelect, inaert, update, delete
- .. u.•
to ..11a1localboat identified by 'paaavord'1
```


Recuerde que puede ejecutar e;ste SQL si escribe

```
ayaql -u root ·Pe create_databaae.aql
```

Tendr6 que introducir su contrasefta de usuario nfz. También debería cambiar la contraseiia del usuario de correo en create_database ·sql y en db_fns. pbp antes de ejecutarlo.

En el CD-ROM se incluye un archivo SQL con el nombre populate. sql. En esta aplicación. no vamos a crear un proceso de registro o de administraci6n de usuarios. Puede ailadir uno personalmente si quiere utilizar este software a mayor escala, pero si sólo lo quiere para uso personal, basta con recurrir a la base de datos. La secuencia de comandos populate. sql proporciona un modelo para hacerlo, por lo que sólo tiene que introducir sus detalles y ejecutarla para configurarse como usuario.

Arquitectura de la secuencia de comandos

Como hemos mencionado anteriormente, esta aplicación utiliza ~ secuencia de comandos para controlarlo todo. Esta secuencia se denomina in~. pbp y se incluye en el listado 29.2. Es bastante extensa. pero la analizaremos sec:ci6n a secci6n.

LlatMlo 29.2. Index.php. Parte prlncipsl del sistema Wann Mali.

```
c?pbp
// Bate archivo••  el cuerpo principal de la aplicaci6n Warm Na~.
// Punciona b&aica..nte COII0 un equipo de aatado y aueatra a loa uauarioa
// al reaultado de la acci6n que hayan nleccioaado.

,,''''''
```

```
"!.
        //adj-•   e.ca~      ai bay algo ant:H del encabes.cloque tenga rHultadoa
        Satatu   •  ";

        // pri..ro debe pzac:eaar laa aolicitudea da conexi6D y daaconexi6D
        if ($UHZ"D&MI Spa••-rdl
        (
           if (logiD($UHIC"D&M, $paaawd))
           (
              Satatua •• •cp>Logged in aucceaafully.c/p>cbr />cbr />cbr />cbr />
                 cbr />cbr />';
              S_SUSIOll('autll_uMr'I • SuaeZ"D&M;
              tf(Dumller_of_accounta<S_SESSIOH('auth_uaer'II••ll

                 Saccounta • get_account_liat(S_Sl!SSIOll(•auth_uaer•J);
                 s_sSSSIOlll'••l•ctad_account'l • Saccounta(0J;

           }
           •]••
           (
              Satatua •• •cp>Sorry, we could DOt log youin  witb. that
                         uaerna-  aDd paaaword.c/p><br />cbr />cbr i>cbr />
                         cbr /><br />';

        }
        if(SactiOD • • 'log·out'I
        (
           ••••ion deatroy(I;
           unaet (SactioDI;
           $_S118SION•Hray0l


        lí debe proceaar la aalecci6D, aliainaci6D o al..cenallianto da cuenta•
        /Fet•• de dibu:lar .1· encabesado
        awitcb. < Saction I
        (
           ca•• 'daleta-account•
           (
              delate_account($_SUSIOIII•auth_uaar•J,  Saccountl,
              break;
           }
           ca•• •atora·HttiDga•
           (
              atore_accouat_aettinga(S_SESSIOll['auth_uaer•J, S_POSTI;
              break,

           caH '••lect·account•
           (
              II ai baa aeleccioaado una cuenta v6lida, al..c6Dela –    variable
              // da Hai6D
              if(Saccouat66account_exiata(S_SllSSIOll('autll_uaer•J, Saccountll
              (
                 S_SIISSIOlll'••lected_account•J• Saccount;


        }
        11 defina  loa botonaa que aparecerá  en la barra de herralliantaa
```



```
        Sbuttona[OJ • •v1......1111ox•;
        Sbutt-111   •  ·–·••aaga•;
        SbuttODa(2) • •account:·••tup•,
        //aol-ta ofrHca  UD bot6D de deaconaxi4n ai HtAD  -tadoa
        if(check_auth_uaer(II
        (
           Sllllt-141 • •log·out';

,,''...''....'''.'.....''''..''.............'.'...'.''.''.''''.'.....'....
// Fue  2: aDCabHadoa
/I IDvle loa encabnadoa  R'1'lIL y la barra da mD4 correapondiante
//ala   acci6D actual
,,
        if(SactiODI
        (
           // 1111..tra el ancabHado  con al -atra  da la aplicaci&a Y la
           /I daacripci6D de la p&giDa o acci4n
           do htal 1leaderlS SIIISSICl11['auth uaar•J, •waza 11ail
              -    -        io~t  action(Saction1,
                            s_ssssIOlll'••lected_account•JI,
        }
        alN
        (
           // -tra  el anc.,.sado a6lo con al DOllbra da La aplicaci4n
           do ht•l b.eadar(S S11SSIOll('autb. uaar• l, •warm 11ail",
           S_SBSSIOll['aelected_account•J)7


        diaplay_toolbarlSbuttonal;
,,''''''
```

```
    d:laplay_acc__t__Htup <t_sssuom('autb_uaer'J),
    break,

e••• '•end•••nve•
(
    :lf(Hlld_Mange($to, ,ce, ,n11:1act, tM•ngell
      ecbo •cp>illeaaag9 Hat.c/p><br /><br /><br /><br />cbr
      />cbr />'1
    •1••
      echo 'cp>Could aot ••ad Mnaga.c/p>cbr />cbr />cbr /><br.
      />cbr />cbr />'1
    bnak,

e••• 'delata• ,
(
    dalat•—••nve 1,_1us1m ['autb_uaar' J,
        ,_sus1a111•aalactad_accouat'J, •••av•:41,
    // UjaH qua ao bay 'Alto' • coatiaua— coa al a:lgu:1—ta caao

ca•• 'aalact·acc—t'
e••• 'viaw- ..ilbox' ,
(
    // d H Hlacc:loaa uilbox o v:lav ..ubolc, ••tn el bua6a
    diaplay_l:latl,_SUIIOll[•autb_uaar'J,
      ,_sUSIOII['Hlacted_accouat'|1|
    break,

ca•• '•bov·baadara' ,
ca•• 'bida·baaclara'
ca•• •viav·..••av•'
(
    // a:l acaba—• ele Hlacc:Loaar UD •an:I• da la l:l.ata, o Hta-•
    // -.1:laaado UD ..a••:1• y apt.—a por ocultar o ver loa
    // aa.cabHadoa, cargue UD —•:1•
    •fullbaadar• • 1,actioa• •'•bow-baaclara•1,
    d:laplay_....ave!,_1u11a11[•autb_uaar'J,
                ,_susIOlll'••lacted_a ~~ t•J,
                •••aga:ld, •fullbaadaral,
    bnak1
)
caaa 'nply-all• ,
(
    //aatablaaca ce~ la liaea ce antigua
    :lfll'1•PI
      ,:1.map • opea._.. :llbolcl•_ !Cll('autb_UHr'J,
                ,_ !Cll('fflacted_accouat'J1,
    if!Uup)
    (
      •baadar • :lup_baadar(tiup, —••age:ldl|
      :lf(,baadar->nplIy_toaddraaal
        Sto• •baadar->nply_toaddreaa,
      .1 ..
        •to• •baadar·>fZ'Oll&ddreaa,
      ,ce• ,11aacle••>ccaddraa•,
      ••ub:t•ct • 'a., •.,baadar->aubjact;
```



```
      •bodr • ... _... illf(:1.Mp_bo4r($:1.ap, tM•aaga:ld) 1,
      :lup_cloH(•illap~,

      diaplay_aaw_.....fozal,_8818I09('aatb_aaar•J,
      tto, ,ce, ••ubject, tbody)/
    )
    brHk,
}
case 'reply'
{
    //aatablaaca la d:lncc:16a ~ nply•to o fro11 del Ma.aaja actual
    :lf 11,:1.iup)
      ,:l.119l> • apaa_.. :l111ox1,_ ...1Im(•autb_uaar'],
                ,_ss111m[•••lected_aceCNDt'JI,
    :lflti•pl
    (
      ,11aa11ar • :1119l>_baadarl$:l ..p, fM•aageid),
      :lfltbaadar->raply_toaddraa•)
        •to• tbaadar->raply_toaddn••1
      •1••
        •to• •baadar·>fZ'Oll&ddraaa,
      •aubjact • 'a., '.,baadar->nl1jact,
      •bodr • add_.-ot:l..(atr:lpalaabaa(:l ..p_bodyl•:l—p,
        tMangeid))lr
      :l•p_cloH<t:l•p) 1
      d:laplay_aav_....aga_fora(,_IUII011['autb_aHr'J,
                •to, ,ce, ,n11:tact, •bodr)/
}
ca•• 'fo~ ·  ,
(
    //aatablaaca al ..aa•:t• ~ cuerpo del 1111l••:t• actual
    :lf 11$:1.•PI
      •:MP • apaa_•:11bolcl$_8818!011['autb_uaar•J,
                -,_1S18I0111'fflacted_ace—t1J)1
    :lflU•pt
    (
      •baadar :1•p_baadar(•:l•p, •—•a&ge:ld)1
      tbody • add_quot:1119(atr:lpèlaallaaCi ..p_bodyC•iup,
        •—••ave:l.41111
      ,n11:1act • 'Fl,d, '.,ballllar·>nl1:1•ct1
      :l..p_cloHC•i•p)1
      d:laplay_aav_...aaga_foza(,_1881I011['autb_uaar•J,
                •to, ,ce, ,n11:1act, •body1,
    )
    bnalt|
)
e••• '——''ve' ,
    diaplay_....,_...aage_fozal,_1 ...!011['autb_uaar•J,
                •to, ,ce, ••ub:tact, tbody)1
    break,
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
   }
,,...'''.............'.....................................'.....'.....''.....
// ,... ••pie·
,,                                                                          '
   do_htal_footer(l1
?>
```

Esta secuencia de comandos utiliza un enfoque de control de eventos. Contiene la lógica para determinar qw! función hay que invocar para cada evento. En este caso, los eventos se desencadenan cuando el usuario pulsa.alguno de los botones del sitio, cada uno de los cuales selecciona una acción. La mayoría de los botones se generan por medio de la función `display_button` (l pero, si se trata de un botón de envio, se utiliza la función `display_form_button()`. Ambas funciones se encuentran en `output_fns`. php. Todas acceden a URL con la siguiente forma

```
index.php?action•log-ouc
```

El valor de la variable `action` cuando se invoca index.php demrmina qw! controlador de evento se activa. Las cuatro principales secciones de la secuencia de comandos son las siguientes:

1. Es necesario realizar algún procesamiento antes de enviar el encabezado de p,gina al navegador, como por ejemplo iniciar la sesión, ejecutar cualquier procesamiento previo para la acción seleccionada por el usuario *y* decidir el aspecto de los encabezados.
2. Procesamos *y* enviamos los correspondientes encabezados *y* la barra de menú para la acción seleccionada por el usuario;
3. Seleccionamos el cuerpo de la secuencia de comandos que se va a ejecutar, en función de la acción seleccionada. Las distintas acciones desencadenan diferentes llamadas de funciones.
4. Enviamos los pies de p¸gina.

Si observa el código de la secuencia de comandos, comprobar, que estas cuatro secciones est,n marcadas con comentarios. Para comprender cómo funciona este sitio, describiremos el sitio acción por acción.

Iniciar y cerrar sesión

Cuando un usuario carga la p'gina `index`. php, accede al resultado que mostramos en la figura 29.2.

éste es el comportamiento predeterminado de la aplicación. Como todavía no hemos seleccionado ninguna $act ion *y* no hemos introducido detalles de conexión, ejecutaremos las siguientes partes del código.

Flguni 21.2. La pantalla de inicio de sesión de Warm Mal le padH III nombre de usuario y IU conlraAfta.

En la fase de preprocesamiento,en primer lugar ejecutamos el siguiente código:

```
include ('include_fn•.pbp'I;
••••ion_•tart(I;
```

Estas lineas inician la sesión que se utilizan para realizar el seguimiento de las variables de sesión $auth_user *y* $selected_account, que veremos m's adelante. Al igual que en el resto de aplicaciones, crearemos nombres de variables cortos. Lo hemos hecho en todas las secuencias de comandos relacionadas desde el primer capítulo, por lo que no merece la pena mencionarlo, excepto para la variable action. Por ello, lo extraemos de la matriz $_RBQUESTI endremos que hacer lo mismo con la variable account, a la que normalmente se-accede a través de GBT, pero con la que se utiliza POSTpara eliminar una cuenta. Para ahorrar trabajo a la hora de personalizar la interfaz de usuario, los botones que aparecen en la barra de herramientas los controla una matriz. Declaramos una matriz vada

```
Sbuttons • array(l1
```

y definimos los botones que queremos incluir en la p¸gina:

```
Sbutton•[0J  •  •vtev- ..ilbox';
Sbutton•[ll  •  'nev- ..••age•,
Sbutton•[2]  •  •accounc-eeeup•;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Si el ususios inicia sesión posteriormentecomQadministrador,aftadiremosm4s botones a esta matriz. En la fase del encabezado, imprimimos un encabezado normal de color vainilla:

```
do_htlll_bellderC$_8SUIOll['auth_uaer•J, •vara Nail',
      s_ssaSIOll['..laotad_acc:ount•J!
cliaplay_toolbarC$buttoaai
```

Este código imprime la barra de título y de encabezado y, tras ello, la barra de herramientas de botones que se puede ver en la figura 29.2. Estas funciones se encuentranen la biblioteca de funciones output _fna .php, pero como su efecto se puede comprobar en la imagen, no las analizaremos aquf. Seguidamente llegamos al cuerpo del código:

```
ifCobaok _auth_u..r(II
(
   aoho 'cp>YOU naad to 109 in•1
   if($aotion••S•otion1•'log-out'I
      aobo ' to go to '.fonat_aoti—($aotionl
   echo '.c/p>cbr />cbr />'1
   diaplay_logia_form($aotionl1
```

La función check_auth_uaer () proviene de la bibliotecauaer_auth_fna.php. Hemos utilizado un código similar en algunos de los proyectos anteriores, para comprobar si el usuario est4 conectado. Si no lo está, como sucede en este caso, le mostramos un formulario de inicio de sesión, que puede ver en la figura 29.2. Este formulario se dibuja en la función display_login_form() deoutput_fna.php.

Si el ususio completa el formulario correctamente y pulsa el botón Lag In, verá el resultado que se muestra en la figura 29.3.

En esta ejecución de la secuencia de comandos, activaremos distintas secciones del c6digo. El formulario de inicio de sesión tiene dos campos, $uaemame y $pasaword. Si estos campos se han completado, se activará el siguiente segmento de código de preprocesamiento:

```
if ($uaenua ll Spaaaword1
(
   if llogia(Sua•r—.     $paaawdl1
   (
      $atatua '•  •cp>Logged ia auooaaafully.c/p>cbr />cbr />cbr />cbr />
            cbr />cbr />'1
      $_8888IOll['auth_uaer•J • $uaanua1
      if(nllllber_of_aoooata($_SBSSIOll[•autb_uaar•JI• •II
      (
         $aoooata • get_accouat_liat($_8888IOll['autb_uaar'Jl1
         $_8B88IOll['aalected_aocouat•J • $aoooata[OlI
   )
   •|••
   (
      $atatua '•  '<P>lorry, ve could not log you in witb that
```



```
uaaraaaie aad paaaword.c/p>cbr />Ctlr />cbr />cbr />cbr />cbr />'1
>.
```

Figura 29.3. DNpués de conec:tanNI satlsf8CIDrlllm. el usuario puede empeur a utilizar la epllcacl6n.

Como puede comprobar, el código invoca la función login (1, similar a la que. hemos utilizado en capítulos anteriores. Si todo funciona co~te. registramos el nombre de usuario en la variable de sesión auth uHr.

Adem4s de configurar los botones que vemos cuando no estamos conectados, aftadiremos otro botón para pennitir que el usuario pueda desconectarse de nuevo, como se indica a continuación:

```
if(obaok_autb_uaarCII
(
      $buttona[4] • 'log-out'1
```

Puede ver el botón Lag Oat en la figura 29.3. En la fase del encabezado, mostramos de nuevo el encabezado y los botones. En el cuerpo, mostramos el mensaje de estado que definimos anteriormente:

```
aobo $atatua,
```

Tras ello, basta con imprimir el pie y esperar a ver qué hace el ususio.

EL_HOLISTICO@HOTMAIL.COM - 2006

Configurar cuentas

Cuando un usuario utiliza por primera vez el sistema Warm Mail, tendrá que configurar alguna cuenta de correo. Si el usuario pulsa el botón Accoant Setap, se define la variable action como account-setup y se vuelve a invocar la secuencia de comandos index. pbp. Tras ello, el usuario verá el resultado que mostramos en la figura 29.4.

Figura 21.4. El usuario tiene que configurar su cuenta de correo electrónico antes de que pueda leer su correo.

Volvamos a la secuencia de comandos del listado 29.2. En esta ocasión, debido al valor de $action, obtendremos un comportamiento diferente. El encabezado obtenido es distinto, como se indica a continuación:

```
do htal beader($ Sl!SSIOll[•auth uaer'I, 'wara Nail - '.
  -    -    foraat_aétion($action1,   $_SBSSIOll[¡aelectad_account'))¡
```

. Lo que es rús importante, obtenemos un cuerpo düerente:

```
ca•• 'atore-aettinga•   ,
ca.. 'account-•etup•    :
ca.. ¡delete-account•   ;
(
```

```
diaplay_account_aetup(S_SE88IOll('auth_uaer'll;
break;
```

Es el modelo habitual: cada comando invoca una fundón. En este caso, invocamos la funci6ndisplay_account_setup (), cuyocddigo se muestra enel listado29.3.

Listado 29.3. Función dlsplay_account_setup() de output_fns.php. Función para obtener y mostrar detalles de las cuentas.

```
function diaplay_account_aetup($auth_uaerl
(
   //aueatre un foraulerio •new eccount• vaclo

   diaplay_account_fore($auth_uaerll
   $11at • get_accounta|Sauth_uaerl;
   $accounte • aizeof($11atl;

   // 1111eatre toda• laa cuenta• al..cenadaa
   foreach($11et ae $key •> Saccountl
   f
      // mueetre un forl111lario para loe detalle• de cada CIMllta.
      // fljeee en que vaaoe a enviar la ~ traeefta de toda• lae cuenta• en
      // el IITNL no ee aconeejable
      dieplay_account_fore|Sauth_ueer,  $accouilt[•accountid•J.
                              $account ('eerver•) ,$aCCOIIDt(' ~ eUHr'J,
                              $eccouat('re110tepaeeword'J,$account('tJ1!8'l,
                              $account (¡port'11]
```

Al invocar esta función, muestra un formulario en blanco para aftadir una nueva cuenta, seguido por formularios modificables que contienen cada una de las cuentas de correo actuales del usuario. La función display account form () mostrará el formulario de la figura 29.4. Comprobará que lo~tilizamosde dos formas diferentes. Lo utilizamos sin parámetros para mostrar un formulario vado y lo utilizamos con un conjunto de parámetros para representar un registro existente. Esta función se encuentra en la biblioteca output_fns. pbp y simplemente representa código HTML, por lo que no la analizaremos.

La función que recupera todas las cuentas existentes es get _accounts (), de la biblioteca __il_fns. pbp. En el listado 29.4se incluye esta función.

._..... 21.4. Función get_accounts() de rmal1_rns.p11p. Función para rec:upersIDdos 1os detalles de cuentas de un determinado usuario.

```
¡uaction get_accounta(Sauth_uaerl

   $11at • array(I;
   if($conn•db _connectCII
   (
      $query • •eelect • froa accounta wbere uaernaae.  'S•!1th_u•er••l
      $result • Scomt->query|$queryl ¡
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
    if1$nH1t)
    (
      wbile($••ttill!f• • $re•ult•>fetch a••oc(II
        array_pli•bl $li•t, $••ttiag•I;
    •h•
      return f•l••,
  return $lht1
```

Como puede comprobar, esta función se conecta a la base de datos, reaapera todu las cuentas de un determinado usuario y los devuelve en forma de matriz.

Crear una nueva cuenta

Si un usuario completa el formulario de cuentas y pulsa el botón S.ve <ltanpl, se activa la acción atore- set tinge. A continuación veremos el c6digo de control de eventos para ello desde index. php. En la fase de preproc:esamiento, ejecutamos el siguiente c6digo:

```
ca•• '•tor•···tting•'  ,
(
   •tore_account_•etting•1$_88SSJ011(autb_u•er•J,    $_POSTI,
   break,
}.
```

La función atore_account _settings () escribe los detalles de la nueva cuenta en la base de datos. Su c6digo se describe en el listado 29.S.

Umdo 21.5. Funci6n store_account_NltInga()de mall_fns.php. Funcl6n pera guardar nuevos detalles de c:uenlAlS de un UIUMO••

```
function •tore_account_•etting•($autb_u••r,  $•etting•I
(
   if11filled_out1$ ..ttill!f•II
   (
      echo 'All field• 1111•t be filled in.  Try again.cbr />cbr />',
      return fabe,
   }
   •l••
   |
      if($••ttill!J•[•account')>0I
         $query • •upclate account•••t ••rv•r • '$eetti119•C••rver)•,
               port • $••ttill9•lportl,  type • •$..ttinge[type)•,
               r111111teu••r • ;$•etting•(r111111teu•er)•,
               remotepa••-rd • '$••tting•[remotepu.-rd)'
            wbere accountid • $••tti119e(account)
               IIDd u..m&1111 • '$autb_u••r•,
      'h•
         $query • •ineert into account• valuee ('Sauth u..r•,
               '$•ettiag•t••rver)•, $••ttingelport),   -
```



```
u,,  ___ ___ coaaactCJ>
(
   $re•ult•$COIID•>qaery(lqlleryl r
   i'f ($renlt)
      return true,
   •l••
      return falN1
)
•l••
(
   ecbo 'could not etore cllIIIgM.cllr />en />cbr />cbr />cbr /Mbr ¡.,.,,
   return f•l••'
```

Como puede apreciar, dol de las opcianes de esta función permiten aftadir una nueva cuenta o actualizar una cuenta ya existente. La fund6n ejecuta la couespondiente consulta para guardar las detalles de la cuenta. Despuá de almac:ienaños detalles de la cuenta, vohemos a illdex. pbp, a la fase del cuerpo principal:

```
caee '•tore-eettinge'  ,
ca•• •account···tup•  ,
ca•• 'delete-account•  ,
(
   di9Play_account_••tup($_SSSSI0111'••t1a_.....r1)1
   break,
```

Trasello,ejecutamosla fundóndisplay_account_setup () cori,antespara enumerar los detalles de la cuenta del usuario. Seguidamente, se aAadiñ la nueva cuenta.

Modificar una cuenta existente

m proceso de modific:ad6n de una cuenta existente es muy similar. m usuario puede cambiar los detalles de la cuenta y pulsar el botón Save Owlgel. De nuevo, se desencadenan la acción store-settings pero, en esta ocasión, actualizari los detalles de la cuenta en lugar de aftadirlol.

Eliminar una cuenta

Para eliminar una cuenta, el usuario puede pulsar el botón Delete Aa:oaat que se encuentra debajo de cada listado de cuenta. De esta forma se activa la función delete-account. En la sec:ci6n de preprocesamienlo de la secuencia de comandos index. php, ejecutamos el siguiente código:

```
ca•• 'delete-account•  ,
(
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
clelete_accouacCS_IUllml('autll_llllerJ,        SacCOUDtt,
break;
```

Este código invoca la funcióndelete_account (). Elc6digodeestafunciónse recoge en el listado 29.6. Es necesario controlar la eliminación de cuentas antes del encabezado, ya que la opción de qui cuenta se va a utilizu se encuentra dentro del encabezado. Tambifn sen necesario actualizar la lista de cuentas antes de poderlo ch1,ujar correctamente.

ua.. do 29.1. Funci6n delete_account() de mell_fnl.php. Funqón p.- lllmlMr loa delmlN de Unll cueia.

```
fulic:tion delete_ac:c:ountlSauth_llllerJ  Saccountidl
(
   //elilú.-  una de laa cu-t•• de ••t• lllluario de la ba•• de dato•

   Squery.  •del•t•  froa account• vilera accountid•'S&cCOllllltid;
            ......  118el'Dalle  - 'saut~_....r••,
   if(.CODD•db_c:oaaact(II
   (
      Sreault • Sconn->queryCSqueryl,
   }
   raturn Sraault,
```

Después de que la ejecución vuelve a index. php, la fase del cuerpo ejecuta el siguiente código:

```
c:ase-'atora‰aettiAt•'
ca••  'aCCOUDt-aatup•  :
caH  ,delete-acCOUDt,  :
(
   diaplay_account_aetup(S_SUSlOlll'autb_uaer'II,
   break,
```

Comprobar6que este código ea el mismo que ejecutamos anteriormente; simplemente muestra la lista de las cuentas del usuario.

L~~r correo

Después de que el usuario ha configurado algunas cuentas, podemos pasar a la parte principal, la conexión a dichas cuentas y la lectura del correo.

Seleccionar una cuenta

Tendremos que seleccionar una de las cuentas del usuario para leer el coneo de la misma. La cuenta ac:tualmente selecc:ionada se almacena en la variable de sesión

$selected_account. Si el usuario sólo tiene una cuenta registrada en el sistema. se seleccionar6autom6ticamente cuando se conecte,como se indica a continuadón:

```
iflD.Ullber_of_acCOUDtalS_IUII«m[•autb_uaar•JI•  •11
(
   Sacc:Gllllta.  gec_accounc_liacc•_aaa11a11['aucb_uaar•JI
   s_susIOll['Hlecced_accouac•J  •  Saccounca[OJ,
```

La funciónnUllber_of_accounta (), de__il_fn• .php,se utiliza para determinar si el usuario tiene m'9 de una cuenta. La función get_account_liat () recupera una matriz de los nombres de las cuentu del usuario. En este caso sólo hay una. por lo que podemos acceder a la misma *como* el valor o de la matriz.

La funci6nnumber_of_accounu O se muestra en el listado29.7",

Listado 29.7. Func:16n number_of_accounta de ffllll_fna.php. Func:i6n l*8 d•nnln• el nOmero de cuentlla que ha reglllrado un uauerto.

```
funcciaa llllllller_of_accouncalSaucb_llllert
(
   // obtenga el lldmero de cuentu  de aate lllluario

   Squary • •••lace  countl•I  fram accounta vbere uaenamla • •sautb_uaar••;

   i f ISCODD•db_caaaec:c lll
   {
      Sreaulc • ScGDD·>query(Squeryl,
         if(Sreaulcl
         (
            Srov • Sreault·>fetc~_arrayllr
            return Srov[OJ I


   recum  0,
```

La func:i6n get_account_liat () es similar a la función get_accountll! () que mencionamos anteriormentea excepción de que sólo RCUpera loe nombres lle wmen~

Si un usuario ha registrado varias cuentas, téndr6 que selec:cianar la que quiere utilizar. En este caso, los encabezados contendrin una inatrucci6n SBLBCT que enumera todos los buzones disponibles. Al seleccionar el buzón apropiado,se muestra el correspondiente a esa cuenta, coaio puede •predar en la &gura 29.5.

Esta opción SILBCT se genera en la fund6n do_htal_header (> de output_fn•. php, como se muestra en el siguiente &agmento de código:

```
// incluya el bu:16D da la cuenca Mleccionacla ai el unario  ciene • •
//  de una cu-ca
iflllllllber_of_accounca(Sauth_uaerl>ll
(
   ec:ba  'cf0:nl C&Z99t•"illd8x.pbp?acCi-•opeD•Mi1Jlox•   -bod•ªJI08C">'  ,.
   ec:bo 'ccd bgc:olor•"lfflloO"  allgn••ri9bt•  vallgn••lliddle•>'I
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
dlaplay_account_aelect($auth_uaer,      $aelected_account1,
echo  ıc/td>ıl
echo  'e/forro'  l
```

Figura 21.5. AJ seleccionar la cuenta del equipo local en al cuadro SELECT, se d89CIII'g8 y se mueatra el COffIIO de dlclla cuenta.

Hem.os evitado tener que analizar el HTML utilizado en los ejemplos del libro, pero el que genera la función diaplay _account_aehct ( ) merece una mención especial. En función de Ju cuentas que tenga el usuario actual, dillplay _account _ aelect ( ) generar4 un HTML como el que inostramos a continuadón:

```
caelect oncbange•"wlndow.locatlon•thla.optl0118[aelectedindexJ.value•
        name••accouat•>
   coptlon value••0•  ••lactad>
      Cbooae  Accountc/a>
   coptlon  value•"lndex.php?actlon••elect-account'8ccount•10">
      • U. domalD,COII
   c/optlOD>
   coptlon  value•"lndex.php?actlon•aelect-account'8ccount•11">
     •U  'aerver. coa
   c/optlOD>
   coptlon value•"lndex.pbp?actlon•aelect-account'8ccount•9">
      localboat
   c/optiOD>
c/aelact>
```

La mayor pule de este código es simplemente un elemenlD de selección HTML pero tambiát incluye derta cantidad de JavaScript. Del mismo modo que PHP puede generar HTML, tambiát puede generar secuencias de comandos del lado del cliente.

Siempre que ae produzca un evento de cambio en este elemento, JavaScript define window.location con el valor de la opción. Si el usuario selecciona la primera opci6n.window.location se define como • index. pbp?action•aelect-account1,account•1 O•. Como resultado, se carga este URL Evidentemente,si el navegador del usuario no es compatible con JavaScript o el usuario lo ha desactivado, el códfgo no tenc:lr4 efecto alguno.

La funci6ndiaplay_account_aelect ( ), deoutput_fna .php,seutiliza para obtener la lisia de cuentas disponibles y para mostrar la opción SBLBCT. Tambim utiliza la fundónget_account_liat ( ) que desaibimounterionnente Al seleccionar una de lM opciones de SBLBCT se activa el evento aelect_account. Si se fija en el URL de la figura 29.5, comprobm que se ha adjuntado al final del URL, junto con el Id. de la cuenta seleccionada.

Esto tiene dos efectos. En primer lugar, en la fase de preprocesamiento de index .php, la cuenta seleccionada se almacena en la variable de sesión $aelected_account, como se indica a continuación:

```
ca•• 'aelect·eccount•  :
(
   //el••  ba aeleccionedo una cuenta vi.licia, al•cl.Mla —  - variable de aeai6D
   lf|$account'6account_mclata|$_S888IOJl[~autb_uaar•J,    $account))
   (
      $_S888IClll['aalected_account•J  •  $accqunt;
```

Por otra parte, al ejecutar la fase de cuerpo de la secuencia de comandos; se ejecuta el siguiente código:

```
e•••  ıaelect-eccount ı
c..e  ıvlew ·•ilbax ı
(
   // el ••  aceba de aelecclonar  • Ubox  o vlew ..ubax,  —•tre el bus6D
   diaplay_llat($_SIUISIOll['auth_uaar•J, $_SBSSIOJ1[ıaelected_accountıll;
   break;
```

Como puede comprobar, se realiza la misma acción que si el usuario hubiera seleccionado la opción Vlew Mallbox. Lo veremos a continuación.

Ver los contenidos del buzón de correo

Los contenidos del buzón de correo se pueden ver por medio de la función diaplay_liat ( ). De esta forma se mostrar4 una lista de todos los mensajes del buzón de correo. m código de esta función se incluye en el listado 29.8.

EL_HOLISTICO@HOTMAIL.COM - 2006

U..._ 21.1. Funci6n di9pl8y_Ilt() de oulput_fna.php. Funci6n S-- mostrs lados loa menujN del bumn.

```
f'IIIIC:tioa dl11Play_liat(Sautb_uaar, SaccOUDtidl
(
   // ~r•     la Uata da ——j•• da ••t• bua6D

   global Stabla_widtb1

   u11sac-t1dl
   (
      ac:1lo 'lfo -ilbox   ••lactadcbr /.,.cbr /...cbr /...cbr /.,,,_br /...cbr /.,..':

   . 1..
   (


      ifCSi.apl
      (
         ecbo •ctabla widtl1-Stabla_widtb' callapaclag•'D'
                  callpaddiag•'''  bolrdar:-'D' .,..t

         Sbaadara • iup_baadara(Si-pl;
         // podriamoa volver a aplicar foz:mato a aatoa dato• u obtener otro•
         // detall•• por 11&dio da i—, _fatcllllaadara, paro ca.o ao ••un.al
         // ra9Um&D, ai1pl...nta duplicar..oa cada uno

         ....aagea • aiseof IShaaderal ı
         for|Si  • D; Sics-aaaagea; Si++I
         (
            echo •ccr…ctd bgcolor •••  ,
            UCSit21
               ecbo •tffffff':
            •]••
               echo 'lffffcc•:
            echo ''…ca href••indax. pbp?action•vi--  • • rnve,...aaagaid-
", (Si+1' ''.,.. ı
            echo Sbeadan(Sil,
            aabo •c/a.,.c/t4"'c/tr»\n•1
         }
         ecbo  'e/table»•:
      }
      •]••
      (
         s.ccount • 9et_account_aettin9a(Sauth_uaar, $acC011Dtidl1
         eabo 'c-ld  not open -ū  box '.SaCCOIIDt['Nrvar•J.
            '.cbr  /.,.cbr /.,.cbr /.,.cbr /»';
```

En ella función empezalDOIl a utilizar las funciones IMAP de PHP. Las dos partes principales de esta función son la apertura del correo y la lectura de los encabeDCioe de mensaje.

Para abrir el buz6n de la cuenta de un usuario se invoca la funci6n open_ —111:Jox O que hemos escrito en .. u_fn. .pbp. Esta función se muestra en el listado 29.9.

Uatado 21.t. Fund6rropen_IMIIIC,a() de mall_fna.php. Esta funci6n 1e conec:m al bumn de un UIU8rio.

```
functi911 open_.. ilbox(Sautb_uaar, SacCOUDtidl
(
   // Hlacciona  al bus6n  d  a6lo b&y uno
   tfCnumber_of_accounta(Sautb_uaer)••11

      SacCOUDta • get_account_liat(Sautb_uaerl,
      S_SUSI<m[•aalectad_account•J • SaCCOUDta(OJ;
      Saccountid • sac~ta[oJ,


   11 realice la com&i6n al ••rvidor POP1 o INAP qua baya aalaccioaado al uauario
   S••ttiag• • gat_account_aettillga(Sautb_uaar, Saccountid)1
   ifl|aiseof(Saattiagal)  returl1 o,
   S-ilbox  • '('.SHttinga(Hrftr)   1 ,
   if(Saattinga(typal• •'POP1')
      S-ilbox .•  •/popl'¡
   S.ailbox '• 'ı ·.$aettiaga [port).'}IIIIIOZ';

   // a11111- la ad.,.rtancia y ao olYida cmprobar  el ftlor daYUlto
• Siup  • i•p_open(S ..ilbox, Saettiaga[•r-tauaar•J,
      Sawtti11911('1'9110tepaaaword'J1ı
   I'!lturn   Siup;
```

De hecho hemos abierto el buz6n con la~ 1...:.p_open()~-cp;¡.;;--;¡-- siguienle prototipo:

```
int imap_open (atring bus6a, atriag IJom1lra de uau.rlo, atriag coatr..aa..
[, iDt opc10Daa))
```

Los parámetros que debemos pasar a la función son los siguientes:

- buzón: Esta cadena debe incluir el nombre del servidor y el nombre del buzón y, opcionalmente, un número de puerto y un protocolo. El foñnato de la cadena es

  (110labrdel ••.rYidor/protocolo:puarto}ac.l>re del bus6a

  Si no especificamos un protocolo, se utiliza de forma pndeterminada IMAP. En el código que hemos escrito, comprobari que especificamos POP.3 si el usuario ha especificado dicho protocolo para una determinada cuenta. Por ejemplo, para leer correo electrónico desde el equipo local, utilizando los puertos predeterminados, pochiamc,s. utilizar el siguiente nombie de buzón paralMAP:

  (localboat:143}I1180l1:

EL_HOLISTICO@HOTMAIL.COM - 2006

y este otro para POP3:

```
(localboat/popl:110}IIIIIOX
```

- nOllbre de usuario: El nombre cíe usuario de la cuenta
- contraaefla: La contrasella de la cuenta

Tambwl podemos pasar indicadores opcionales para especificar opciones como ·abrir el buzón en modo de sólo lectura•.

Conviene mencionar que hemos construido la cadena de buzón paso a paso con el operador de concatenación antes de pasarla a imap_open () . Tendrá que prestar especial atención a la forma de construir esta cadena ya que las cadenas que contienen { $ generan problemas en PHP.

La invocación de esta función devuelve un flujo IMAP si el buzón se puede abrir y false si no se puede abrir. Al terminar con un flujo IMAP, se puede cerrar por medio de imap_cloae (flujo_imap). En nuestra función, el flujo IMAP se pasa de nuevo al programa principal. Tras ello utilizamos la función imap_headera () para obtener los encabezados de correo y representarlos:

```
$beadera • i..p_beadera($i..pl,
```

Esta función devuelve información de encabezado para todos los mensajes de correo del buzón al que nos hayamos conectado. La información se devuelve en forma de matriz, una linea por mensaje. No hemos aplicado ningdn formato. Simplemente devuelve una lfnea por mensaje, por lo que si se fija en la figura 295 podral ver el resultado.

Se puede obtener mals información sobre encabezados de correo si utiliza imap_header (), del mismo nombre pero confusa. En este caso, la función imap_headera () nos ofrece suficiente información para nuestros objetivos.

Leer un mensaje de correo

Hemos configurado todos los mensajes de la función diapi.ay_liat () anterior para vincularlos a determinados mensajes de correo electrónico. Cada uno de estos vínculos tiene la siguiente forma:

```
index.pbp?action•view-..aaage...aaageid•6
```

measage id es el nwnero de secuencia utilizado en los encabezados que recuperamos anteriormente. Comprobará que los mensajes IMAP estaln numerados a partir del 1, no del 0. Si el usuario pulsa sobre uno de estos enlaces, veril el resultado que mostramos en la figura 29.6.

Al introd udr estos paralmetros en la secuencia de comandos index. php, ejecutamos el siguiente código:

```
caM  •abow-beadera•
e••• 'bide-beadera•
```



```
cue  •view-..aaage•  ,
(!
   // ai acab-•  de obtener IIII -:1•  de la Uata o ••t-   anaUHlldo
   // un .. au:l•  y opt-•  por oaaltar o ver eacabeudoll, caqae  un -.1•
... lfullbeadera •  (laction• ••ldlalr•Madera •l,
   diaplay_..aaage(I_SB88I011(ı..  tb_ueer•J,
                    $_8SSSI011['..1ectae1_accouat•J,
                    ,..aaageid, $f•lli-.deral,
   break;
```

Índice de contenidos 13

Figul'II 21.1. Al utllizllr la acción vtew-meuaoe podemoe ver un detennilldo menuJe, en esta cuo, un correo bSIIUl'II.

Apreciará que se comprueba si el valor de $action es igual a •ahow-headera'. En este caso, seral false, y $fullheadera también se definirá como false. Describiremos la acción •ahov-headera • en breve.

La línea

```
$fullbeader• •  ($action• •'•b— ·beadera•),
```

se podría escribir de forma mals clara como

```
if($action•  •'•bow ·beadera•)
   $fullbeadera • true,
eln
   $fullbeadera • falae,
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Tras ello, invocamos la función display _meHage () . Búicamente representa código HTML sencillo, por lo que no la describiremos. Invoca la función retrieve_meaaage () para obtener el correspondiente mensaje del buzón:

```
$..••av• • retr1..._.....9e($auth_u••r, $accountid, S..•••g•id,
$fullhHdmr•l
```

La fundón retrieve _measage ( ) se encuentra en la biblioteca mail _fna . php. Puede ver su código en el listado 29.10.

Listado 21.11. Fund6n 1'911'1eve_rrmuge() de mall_fnLphp. Epa func:16n recupera un mensaje concreto de un buz6n.

```
fuactioa ratr1..._..••age($auth_u••r, $accouatid, $..••av•id, $fullheadar•I
(
   $..•••ge • array(I,

   if(ll$auth_u.er •• S..••ageid •• $accouatidll
      retuna fal••,

   $1..p • opea_mailboxC$auth_u.er, $accouatidl1
   ifll$i.ap)
      retu:cn fal••;

   $he•d•r • 1..p_beader($imap, $..•••g•idl;

   if (|$h..d9rl
      retuna fal••;

   $..•••gel•l:locly'| • imap_l:lodyl$i..p, S•••v•idl,
   iU1$—•••iJe['bocly'D
      $N••age[•body'J • "[Thi• -Haga ha• - bocly)\D\D\D\a\a\a•;

   iflSfullheader•I
      S..•.ag•('fullheader••J • imap_fetchheader($imap, S..•••geid);
   •|••
      S••••ge[•fullbeader•'| • ••;

   S——••9•['91111:lect'l • $header->9Ubject;
   S.... . , . (•f~re ..•J •$header->f~n••;
   S••8a'J9('toaddre••'I • $header->toaddr•••;
   s..••av•|'ccaddr•••'I • $header->ccaddre••;
   •—•••ge('date'I • $header->date;

   JI fl:I••• ea que poclet10• obtener iaformaci6D 116• detallada si utiliza.oe
   11 fr- y to ea lugar de fr-ddre•• y toacldrue, pero ••to• •oa mA•
   JI •eacillo•

   imap_clo•e($i•pl1
   retara S.. •ságe
```

Como en el caso anterior, hemos utilizado open_mailbox () para abrir el buzón del usuario.

Sin embargo, en esta ocasión, buscamos un mensaje concreto. Por medio de esta biblioteca de funciones descargamos de forma separada los encabezados del mensaje y el cuerpo del mismo.

Las tres funciones IMAP que hemos utilizado son imap_header(), imap_fetchheader ( ) e imap_body ( ) • Las dos funciones de encabezado son distintas a imap_headera ( ) , la que utilizamos previamente. Los nombres pueden resul•le confusos. En resumen:

- imap_headers O: Devuelve un resumende los encabezados de todos los mensajes de un buzón. Los devuelve en forma de matriz con un elemento por menaje.
- iMp_header (·, : Devuelve los encabezados de un mensaje conaeto en forma de objeto.
- iaiap_fetchheader (): Devuelve los encabezados de un mensaje concreto en forma de cadena.

En este caso utilizamos imap_header (› para completar campos de encabezado específicos e imap_header () para mostrar los encabezados coJDpletosal usua.-rio en casode que lo solicite (como veremos en breve).

Utilizamos imap_header () e imap_body() para generar una matriz que contenga todos los elementos del mensaje en el que estemos interesados. Invocamos imap_header () de la siguiente forma:

```
Sbeader • 1..p_beaderl$imap, s-••avei4J;
```

Tras ello podemos extraer del objeto todos los campos que nos interesen:

```
S•••age[••ub:lect•J • $header->9Ubject;
```

Invocamos imap_body ( ) para ai\adir el cuerpo del mensaje a la matriz:

```
S..H•p('body' I • imap_bocly(Si..p, S•••ngeidl
```

Por último, cerramos el buzón con imap_close () y devolvemos la matriz que hemos generado. De esta forma, la función diaplay _mesaage() puede mostrar los campos del mensaje como se indica en la figura 29.6.

Ver encabezados de mensaje

Como puede comprobar en la figura 29.6, hay un botón Show Headen. Este botón activa la opción show-headers, que aftade todos los encabezados de mensaje a la pantalla del mensaje. Si el usuario pulsa este botón, verá un resultado similar al de la figura 29.7.

Como habrá comprobado, el control de eventos de view-measage también engloba show-headers (y su antónimo hide-headers). Si se selecciona está opción, realizamos las mismas acciones que en casos anteriores Pero en retrieve-

EL_HOLISTICO@HOTMAIL.COM - 2006

meHagea () tambió obtenemos el texto completo de los encabezados como se indica a continuación:

```
if(Sfullheader•I
      s-Hage l' fullheader•' J • iup_fetchheader C$1up, s-•ngeidl
```

Tras ello podemos mostrar estos encabezados al usuario.

Figura 29.7. El uso de ihow-headers para verloa encabezados completos dente mensaje parmite al usuario localizar el origen del correo basura

Eliminar correo

Si un usuario pulsa el botón Delete de un determinado correo electrónico activará la acción 'dele te', que ejecuta el siguiente código de index. pbp:

```
caae 'delete' :
{
   delete_...•••velS_ssss10N(•autb_u•er•J,
                    s_sBSSION['••lected_account') , S•••~id) ;
   //Uje•e en que no hay '•alto' - continuaremo• con el dguiante ca•o

e••• '•elect-account•
caae •view-mailbox' ,
{
```



```
   // d •e ••lecci-  uilllox o view uilllaK, -tn   el bus6la
   di•play li•tCS SSISICll('auth ..er•I,
        S_Sllssnar1• ..lected~accouat'J11
   bnu;
```

Como puede comprobar, el menuje se borra por medio de la función del!te_mHeage() y, tras ello, se muestra el buzón resultante como mencionamos antmonnenle. El c6cligo de la función delete_-aaage (> se muestra en el listado 29.11.

Uatado 29.11. Funcl6n dlllete_meuaee de mall_n.php.E• funcl6n boml un menuie cancntto de un buZón.

```
(unction delete_-••111•l$nth_ueer, Saccouatid, S••N99_1d)

   // borre un eolo ..n•aje del •ervidor

   Si..p • open_u1lboxl$nth_ueer, Saccountidl;
   UCSiupl
   {
      i..p_delete($iup, S--••ege_idl1
      1map_mcpungel$i..pl1
      iup_cloee(Siupl;
      return true;

   retum fal•1
```

Como puede apreciar, esta función utiliza una serie de funciones IMAP. l,.as nuevusonimap_delete() eimap_expunge().i•p_deleté"t) solamente marca mensajes para su eliminación. Puede marcar todos los mensajes que desee. La invocación de imap_expunge() elimina realmente los mensajes.

Enviar correo

Por último llegamos a la parte de envfo del correo. Por medio de esta secuencia de comandos, el envfo se puede realizar de varias formas: el usuario puede enviar un nuevo mensaje, contestarlo o reenviarlo. Veamos ~o funciona.

Enviar un nuevo mensaje

El usuario puede seleccionar esta opción si pulsa el botón New Meaaage. Se activará la acción 'new-message', que ejecuta el siguiente código en index. pbp:

```
case 'new-message' :
{
   di•play_new_..•••~-fora(S_SSSSION('autb_u•er•J.
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
            $to, $ce, $eubject, $body);
break,
```

El formulario de nuevo mensaje es simplemente un formulario para enviar correo. Puede ver su aspecto en la figura 29.8. En realidad, en esta figura se muestra el reenvío de un mensaje, no el envío de un nuevo correo, pero el formulario es el mismo. En apartados posteriores veremos cómo se responde y se reenvía correo.

Flgur11 29.1. Al utilizar el reenvlo de correo podemos informar del correo basura.

Al pulsar el botón Send MeeNge se invoca la acción "send-meesage~ y se ejecuta el siguiente código:

```
caee •eend-..aeage• :
{
   ifl•end_me•••gel$to, $ce, $eubject, $me•••g•I>
      ecbo •cp>N•••ag• sent.c/p>cbr />cbr />cbr />cbr />cbr />cbr />";
   el••
   ecbo •cp>Could DOt send .....ga.c/p>cbr />cbr />cbr />cbr />cbr }>cbr />";
   break;
```

Este código invoca la función send_message (), que se encarga de enviar el correo. El código de esta función se recoge en el listado 29.12.

U.tado 29.12. Funct6n send_mesuge() de mall_fns.php. Esta función envfa el mensaje que haya escrito el usuario.

```
functioa •end_..... 9a($to, Scc, S•ubject, Seeeeagel
{
   //envle un correo a travf• de PBP

   if !t$conn•db connectlll
   {
      return fal••:
   }
   $query • '•elect addr••• fro• u••r• where '
            .u•erna ..•\''.$_SESSIO•l•auth_u••r•J.•••:

   $re•ult • Sconn->query(Squery);
   if !l $reeult)
   {
      return fahe:
   }
   el•• if ($re•ult~>nua_row•• •01
   {
      return fal••:

   el••
   {
      $row • Sresult->fetch object(l;
      $other • 'Proa: '.$r~ ·>addre••:
      re l!a.pty(Sccll
         $oeller.w•\r\n<:co scc•:
      if !..111sco, S•ubject. S.. •••ge, Sotherll
      {
         return true
      }
      •h•
      {
         return fal••;
```

Como puede comprobar, esta función utiliza mail l) para enviar el correo electrónico. Sin embargo, primero carga la dirección de correo del usu~o desde la base de datos para utilizarla en el campo From del mensaje.

Responder o reenviar correo

Las funciones Reply, Reply Ali y Forward envían correo de la Inisma forma que New Message. La diferencia de funcionamiento radica en que completan Jas partes del formulario de nuevo mensaje antes de mostrárselo al usuario. Volvamos a la figura 29.8. El mensaje que vamos a reenviar se sangra con el símbolo > y a la linea Subject se le ha aftat;(idoTo como prefijo. Del mismo modo, las opciones Reply y~ Reply Ali se completan con los datos de los receptores, la línea del asunto y el

EL_HOLISTICO@HOTMAIL.COM - 2006

mensaje. m c6digo para realizar esta operación se activa en la secd6n del qaerpo de index. php, como se indica a continuación:

```
ca•• ıreply·allı
(
   //aatablaaca ce coao ita.a ce antigua
   if(1 $illap)
      U•p  • opan_•ilbox($_SU8I0ll[•auth_uaer•],
                          $_8U8I011[ı .. 1ectad_accountı]);
   iflti•PI
   (
      $haader • i•p_haadar($i ..p, •-•aageid)ı
      if($header·>reply_toaddraaa)
         $to•  $haad•r·>reply_toaddraaa,
      •]••
         $to•  $haader·>fro1111ddreaa1
      $ce•  $haader·>ccaddreaa,
      $aubject  •  'a.,  '.$headar·>aubject1
      $body • add_quoting(atripala•ha•Cimap_body1$imap, •—••agaidll);
      1.. p_cloaa($i ..pl1

      diaplay_new_..•aaga_forl1($_8U8I011[ıauth_u..r•],
      $to, $ce, $aubjact, $body);


ca•• •raply•  ,
(
   //defina addr••• COIIO reply·to o fro.i del ..naaja actual
   ifl1$imap)
      $1..p • opan_.. ilboxl$_8BS8I0ll[•auth_uaer•],
                          $_8B88I011[•aelected_accountı])1
   if($1Mp)
   (
      $header • iMp_haadar($i..p, ,—aaageid)ı
      if($haadar·>reply_toaddreaa)
         $to•  $haadar->reply_toaddreaa,
      •]••
         $to•  $ha°adar·>fro1111ddreaa;              ...
      $aubject • 'a.,  '.$header·>aubject;
      $body • add_quoting(atripal..hea(i..p_body($iap, ,..aaagaid)II;
      i1lap_e10..($i..pl1

      dlaplay_naw_..aaage_forl1($_8U8I0ll(ıauth_uaer•J,
                 $to, $ce, $aubjaet, $body);


   break,
)
eaaa 'forward•  ,
(
   //defina al ..naaja co.io cuerpo del ..uaja  actual
   if(l$imap)
      $imap • opan_mailbox($_SBS8I0ll[•auth_uaer•J,
                          $_8U8I011(•..1eetad_account•J);
```



```
lf($i ...)

( $11nder • i • p_haaderÍUMP!  $maaaagaidl;
  $body • add_quoting(atripalaahaa(illap_body($i ..p, .....apid)));
  '8allject • 'Plld•  '.Sbeader·>aubject;
  iallp_el~($i..pl1

  diapley_new_..a.age_forml$_8U8IOII(•auth_uaer•J,
             Sto, S!=c, ~aubjact, $body)1
J
bnakr
```

Puede comprobar que cada una de estas opciones define loe conespondientes encabezacb, aplica un formato en cuo de que sea necesario e invoca la fund6n dbplay _new_-Hage_fo:na() para coafigurar el formulario. Con esto hemos comptet.do la funcionalidadde nuestro lector de correo Web.

Ampliar el proyecto

En este proyecto podemos realizar numerosas ampliaciones ~ras. Puede utilizar como referencia el lector de correo que utilice normalmente, pero entre algunas de las posibles mejoru destacamos las siguientes:

- Permitir que los usuarios se registren en el sitio (puede reutilizar parte del c6digo decapftulos anteriores sobre autenticación y perso~6n de usuarioe);
- Mucho&usuarios tienen m6s de una direuión de correo electr6nico, por ejemplouna personal y otra de trabajo. Al cambiar sus dina:iones de correo almacenadas de la tabla de usuarios a la tabla de cuentas, 1es pennitid utilizar varias direcciones y butarfa con cambiar una pequeña cantidad de c6digo. El formulario de envfo de correo debería incluir un cuadro desplegable para seleccionar la dirección que se va a utilizar.
- Se podría permitir enviar, reabir y ver correo con udúvos adjuntos. Si loe usuarios pudieran enviar archivos adjuntos, tendríamos que incorporar opdones de carga de archivos como las descritas en un capitulo anterior. El envfo de correo con archivos adjuntos se analizan en un capftulo posterior.
- Se podrfan ailadir opciones de libreta de direcciones.
- Se podrfan ai\adir opciones de lectura de noticias de red. La lectura desde un servidor NNTP por medio de fundones IMAP es idmtica a la lectura desde un buzón. Basta con especificar un nwnero de puerto y un ~lo cliferentes en la invocación de imap_open(l. En lugar de asignar un nombre como BANDEJADE ENTRADAa un buzón, se le asigna un grupo de noticias desde el que se realiza la lectura. Podría combinarlo con opciones de generad6n de

EL_HOLISTICO@HOTMAIL.COM - 2006

cadenas del proyecto de uno de los siguientes capítulos para crear un lector de noticias buado en subprocesos Web.

A continuación

En el siguiente capítulo creamnos otro proyecto relacionado con el correo electrónico. En esta ocasión diseftamnos una aplicación que admita el envfo de boletines informativos sobre diferentes **temas a los** usuarios S\IICritos **a** nuestro sitio.

EL_HOLISTICO@HOTMAIL.COM - 2006

Después de diseñar una base de suscriptores a nuestro sitio Web, una opción muy acertada consistirfa en mantenemos en contado con ellos y enviarles un boletfn informativo. En este capitulo implementaremos una interfaz para un gestor de listas de correo (o MLM). Algunos MLM permiten que los suscriptores puedan enviar mensajes a otros suscriptores. Nuestro programa será un sistema de boletines informativos, en el que solamente el administrador podrá enviar mensajes. El nombre de nuestro sistema será Pyramid-MLM.

El sistema será similar a muchos de los que ya existen en el mercado. Para hacerse una idea de lo que pretendemos puede visitar ht tp: / /www. topica. com.

Nuestra aplicación permitirá al administrador crear varias listas de correo y enviar boletines informativos a cada una de ellas individualmente. Esta aplicación utilizará la carga de archivos para que el administrador pueda cargar versiones en texto y HTML de los boletines que haya creado. Esto significa que los administradores pueden utilizar cualquier programa para crear los boletines.

Los usuarios podrán suscribirse a cualquiera de las listas de nuestro sitio e indicar si quieren recibir los boletines informativos en texto o en HTML.

Nos centraremos en los siguientes aspectos:

- Carga de ardüvos con varios archivos
- Archivos adjuntos de correo electrónico codificados con MIME
- Correo electrónico con formato HTML
- Formas de administrar contrasellas de usuario sin interacción de éstos

EL_HOLISTICO@HOTMAIL.COM - 2006

El problema

Queremos diseñar un sistema en línea de creaci6n y envío de boletines informativos. Este sistema permitirá aear y enviar distintos boletines a los usuarios y que los usuarios se susaiban a uno o a varios de estos boletines.

En conaeto, los requisitos de este sistema son los siguientes:

- A los administradores se les debe permitir definir y modificar los boletines informativos.
- A los administradores se les debe permitir enviar boletines en texto y en HTML a todos los susaiptores de una lista de correo.
- A los usuarios se les debe permitir registrarse para utilizar el sitio, así como introducir y modificar sus datos.
- A los usuarios se les debe permitir susaibirse a cualquiera de las listas del sitio.
- A los usuarios se les debe permitir anular la susaipción a las listas a las que estén susaitos.
- A los usuarios se les debe permitir almacenar su preferencia con respecto a boletines informativos en formato HTML o en texto.
- Pormotivos de segwidad, a los asuuius nose les debe pe,udth euviat correo a las listas o ver las direcciones de correo electrónico de otros usuarios.
- A los usuarios y administradores se les debe permitir ver información sobre las listas de correo.
- A los usuarios y administradores se les debe permitir ver boletines informativos atrasados que se hayan enviado a una lista (el archivo).

Componentes de la solución

Para cumplir los requisitos anteriores, necesitaremos una serie de componentes. Los principales consisten en definir una base de datos de listas, susaiptores y boletines informativos archivados; la carga de los boletines aeados fuera del sistema en línea y el envfo de correo con archivos adjuntos.

Definir una base de datos de listas y suscriptores

Realizaremos el seguimiento del nombre de usuario y la contraseña de todos los usuarios del sistema, así como de una lista de las listas a las que se hayan susaito. También almacenaremos la preferencia de cada usuario para recibir correo electrónico en HTML o en texto, para poder enviarle la venión adecuada del boletín informativo.

Un administrador será un usuario especializado con capacidad para aear nuevas listas de correo y para enviar boletines informativos a dichas listas.

Seria aconsejable dotar al sistema de un archivo de boletines informativos anteriores. Puede que los suscriptores no guarden boletines pasados pero que les interese buscar algo en conaeto. También se puede utilizar el archivo como herramienta de marketing de los boletines ya que los susaiptores potenciales pueden ver el aspecto de los mismos.

La definici6n de una base de datos en MySQL y la creación de una interfaz para la misma en PHP no resultará difícil ni es nada nuevo.

Cargar archivos

Necesitamos una interfaz para permitir al administrador enviar boletines de noticias, como indicamos previamente. Sin embargo no hemos mencionado cómo los crearán. Podríamos proporcionarles un formulario en el que puedan reda~r o pegar el contenido del boletín informativo. No obstante, si permitimos que los administradores aeen el boletín en su editor de texto favorito y que carguen el archivo al servidor Web, aumentaremos el atractivo del sistema.

Para ello podemos utilizar las opciones de carga de archivos que analizamos en un capítulo anterior.

Será necesario utilizarmdonualuio más coa,plicado que el que helll08 emplea do previamente. El administrador tendrá que carga~veisiones del ~l~ info~ativo tanto en texto como en HTML, junto con las imágenes que quiera mcluir en HTML

Una vez cargado el boletín, tendremos que crear una interfaz para que el administrador pueda ver una vísta prevía del mismo antes de enviarlo. De esta forma. puede confirmar que todos los archivos se cargan c~rrectamente:

Debe saber que almacenaremos todos estos archivos en un d1rectono para que los usuarios puedan leer ejemplares atrasados de los boletines informativos. Es necesario que el usuario bajo el que se ejecute su servidor Web pueda esaibir en este archivo. La secuencia de comandos de carga intentará escribir los boletines informativos en . /archivo/ por lo que debe crear este directorio y definir correctamente los permisos del mismo.

Enviar correo con archivos adjuntos

En este proyecto, queremos envíar a los usuarios ID\ boletín informativo en texto sencillo o en una "atractiva" versión en HTML, en función de lo que prpieran.

Para envíar un archivo HTML con imágenes incrustadas, tendremos que buscar la forma de mandar archivos adjuntos. La sencilla funci6n mail O de PHP no admite el envío de archivos adjuntos. En su lugar, utilizamnos el magnffico paquete Mail _Mimede PEAR, aeado por Richard Heyes, que permite procesar archivos

EL_HOLISTICO@HOTMAIL.COM - 2006

adjuntos HlML y también se puede utilizar para adjuntar cualquier imagen incluida en el archivo HIML. Las instrucciones de instalación de este paquete se incluyen en el apmdice A.

Presentar la selución

En este proyecto, utilizaremos de nuevo un enfoque dirigido por eventos para escribir nuestro c6digo, como hic:imoa en el capítulo anterior.

En primer lugar diseftamos una aerie de diagramas de flujo de sistema para mostrar las rutas que los usuarios pueden tomar a lo largo del sistema. En este caso, hemos diseftado tres diagramas para representar los tres conjuntos de interacciones que los usuarios pueden realizar con el sistema. Los usuarios dispondrm de distintas opciones en función de si no estúl conectados, de si estúl conectados como usuarios convencionales o de si estúl conectados como administradores. En las figuras 30.l, 30.2 y 30.3 se describen cada una de estas acciones.

En la figura 30.l se muestran lu acciones que puede realizar un usuario si no est! conectado. Como puede comprobar, el usuario se puede conectar (si ya dispone de una cuenta), crear una cuenta (en cuo que no tenga una) o ver lu listu de correo a las que se puede suscribir (como t4ctica de marketing).

Figura 30.1. Cuando un uauerto no esté conectado, IOIM!ente puede Mleeelonar un n6mero llmltado de IICClonÑ.

En la figura 30.2 vemos la acción que puede realizar un usuario una vez conectado. Puede cambiar la configuración de su cuenta (direa:ión de correo electrónico y preferenciu), cambiar su contrasei\a y cambiar las listas a lu que est6 suscrito.

En la figura 30.3 se muestran las acciones disponibles cuando se conecta un administrador. Como puede comprobar, un administrador accede pr6cticamente a la misma funcionalidad que un usuario, as(como a opciones adicionales. También puede crear nuevu listas de correo, crear nuevos mensajes para una lista de correo si carga archivos y previsualizar mensajes antes de enviarlos.

Como hemos recurrido a un enfoque dirigido por eventos, la parte principal de la aplicación se incluye en un solo archivo, index. pbp, que invoca un conjunto de bibliotecu de funciones. En la tabla 30.l se incluye un resumen de los archivos de esta aplicación.

Figura 30.2. Une vez conecbldos. las _,... pueden C8fflblenus prefeiencias por medio de distintas apclones.

Figura 30.3. Loa aclmlnlslradores disponen de acciones adic:loMleS.

Tabla 30.1. An:hlvoa de la apllcaci6n de gestl6n de llstas de correo.

Nombre de archivo	Tipo	Descripclón
index.pbp	Apllcaci6n	Sec:uenclll de comandos prtnclpal que ejecuta toda la apllcac:i6n.
include_fne.pbp	Funciones	ColeCCl6n de archivos de Inclual6n de esta aplicllcl6n.
data_valid_fne.pbp	Funciones	colec:d6n de funciones para validar datos lntroduCidOS.
db_fna.pbp	Funciones	Colec:cl6n de funciones par11 realizar la conexi6n a ..... de d8IDS al•.

EL_HOLISTICO@HOTMAIL.COM - 2006

mlm_fn•. php	FuncloQes	Caleccl6n de funciones especifica de esta apllc8cl6n.
output_fn•. php	Funciones	Colección de func:ionn para representar HTML.
upload. php	Componente	Secuencia de comandos que gestiona el componentede carga de archivos del administrador. Eati seearado para facilitar la seguridad.
u•er_auth_fns.php	Funciones	Colecclón de funciones para autenticar usuarios
create_databaH.•ql	SQL	SQLpara definir la base de datos •l• y para configurar un usuario Web y un usu• rio admlnlstrallvo.

A continuaáón describiremos la implementación del proyecto, comenzado con la base de datos en la que almacenaremos información sobre suscriptores y listas.

IÍefinir la base de datos

En esta aplicación, tendremos que almacenar detalles de:

- Liñu: Listas de correo a las que se puede realizar la suscripción.
- Suscriptora: Usuarios del sistema y sus preferencias.
- U.tu de suscriptores: Un registro de qué usuarios se han suscrito a qué listas (una relación varios a varios).
- Correo: Un registro de los mensajes de correo enviados.
- Jmjgenee: Como queremos enviar mensajes de correo electrónico formados por varios archivos (es decir, texto y HTML más una serie de imágenes), también tendremos que saber qué imágenes se incluyen en cada correo electrónico.

El código SQL que hemos escrito para aear esta base de datos se recoge en el listado 30.1.

Llmdo 30.1: create_database.sql. SQLpara crear la base de datos mlm.

```
creace databa•• 111.-,

uae mle!
```



```
create table li•t•
(
   liatid int auto incr...nc not null primary key,
   liatnaee cbar(20l not null,
   blurb varcbar(2SSI
);

create table subacribers
(
   ... 11 cbar(100) noc null primery key,
   realnaee cbarl100) aoc null,
   eiMtype cbar(l) not null,
   pas•-rd cbarltOI not null,
   adlain tinyinc noc null
ll

l aleaceaa una relación entre un suscriptor y una lista
create table aub_liata
(
   ...u cbar(lOOI not null,
   listid int aot null
);

create table eail
(
   eailid int auto increMnt noc null pri..ry key,
   ... u cbar(lOOI-not null,
   subject cbar(lOOI not null,
   Ustid Tnt aoc nulr.
   scacua cbar(10l not null,
   Hnt dateti-,
   IIOdified ti—staep
);

laleacena las ie.6ge-• qae corresponden a un determinado correo
creace table imagea
(
   mailid int not null,
   patb cbar|lOOI not null,
   eiMtype cbar(100l noc null
l;

grant select, illsert, upclate, delete
on ale.•
to •l..localboat identified by •paes~ ';

inaert into aubacriber• valu••
(•adainelocalbost•, •Jldeiniacracive User•, •e•, pea~('adain'I, 11;
```

Recuerde que puede ejecutar este SQLsi escribe

```
aysql -u rooc -p e creace databa••.sql
```

Tendrá que propQrdonar su contrasella de usuario raíz (también podría ejecutar esta secuencia de comandos a través de cualquier usuario de MySQLque tenga los

EL_HOLISTICO@HOTMAIL.COM - 2006

privilegios ~orrectos; simplemente hemos utilizado el usuario rafz por motivos de simplicidad). Debería cambiar la contrasefta del usuario •l• y el administrador de su secuencia de co111Andos antes de ejecutarla. Conviene explicar con mayor detalle alguno de los campos de la base de datos. La•bla liste contiene un li etid y un listname: También contiene un blurb, una descripción de qué es la lista.

La tabla suecribers contiene las direcáones de correo (email) y los nombres (realname) de los suscriptores. También almacena su contraseila (password) y un indicador (admin) para indicar si el usuario es un administrador o no. E:n mimetype almacenaremos también el tipo de correo que prefieren recibir. Puede tratarse de H para HTML o de T para texto.

La tabla sublista contiene direcáones de correo electrónico (email) de la tabla suscribers y listidde la tabla lista.

La tabla mail contiene información sobre todos los mensajes de correo que se envían a través del sistema. Almacena un identificador exclusivo (mailid), la dirección desde la que se envía el correo (email), la lfnea deasunto (subject) y el listid de la lista a la que se ha enviado o la que se va a enviar. El texto o HTML del mensaje puede ser un archivo de gran tamai\o, por lo que almacenaremos el archivo de los mensajes fuera de la base de datos. También realizaremos el seguimiento de la información de estado general: si el mensaje se ha enviado o no (status), cuándo se ha enviado (sent) y una marca de tiempo para mostrar cuándo se ha modificado por última vez (modified).

Por último, utilizamos la tabla images para realizar el seguimiento de todas las

---------t1il\\111&6geeei1111«e!SSHBISS818e!eilla|E1|ilzas&-as-!!!R|Me!!!!!!!!!H|aJNL.e&|—|H'D-tb0b0e!;BIIUHIrIrIVrC9,.. 111e&1ta&1111-iBIIL n.14MgeRrellee&-pl111YeleedlleIIIRII--IlteAmeerlF-IrIHIIIRf-·-----

gran tamaño, por lo que las almacenaremos fuera de la base de datos. En su lugar, utilizaremos el mailid al que estm asociadas, la ruta (path) a la ubicaáón en la que se encuentra almacenada la imagen y el tipo MIME de la imagen (mimetype), por ejemplo, image/gif. El SQL que mostramos anteriormente también define un usuario para que se conecte PHP y un usuario administrativo para el sistema.

Arquitectura de la secuencia de..comandos

Al igual que en el proyecto anterior, hemos empleado un enfoque dirigido por eventos. La parte central de la aplicación es el archivo index. php. Esta secuencia de comandos tiene cuatro segmentos principales, que son los siguientes:

1. Preprocesamiento. Es necesario realizar cualquier procesamiento antes de enviar los encabezados.
2. Configuración y definición de encabezados. Crear y enviar el inicio de la página HTML.
3. Ejecución de la acción. Responder al evento que se ha pasado. Como en el último ejemplo, el evento se incluye en la variable $action.
4. Envío de pies de p,giná.

La práctica totalidad del proa:samiento de la aplicación • lleva a cabo en este archivo. La aplicación también utiliza lu bibliotecude funciones enumeradas en la tabla 30.1, como mencionamos anteriormente.

El c:6digocompleto de la NCUenCia de comandos index.pbp se~ en el listado 30.2.

U•tado 31.2. lndu.php. Archivo prlnc:lpel d8 la aplicac:16n ~~·

```
c?pbp

,
• Secci6n 1:  preproc•••11.1ento
••••••
```

```
!                                                                    ~'.........'
• lacci6D 2: defiaici&I y CODfi:,...i&I de .....                     ~'''''''/
''''''
```

```
        get_real_-c¡gat__  u CI 1, gat_aiMtype Cgat_...u CI 11,.
      break,
    }
    caaa '•bow·otllar·liata•  ,
    {
      diaplay_it... c•Uliliubacrilled Liata',
                    get_UDaubacribed_liataCgat_...11111,  •inforaatiCll1',
                    'ahow,archiva•,   •aubacriba'I,
      break,
    }
    caaa  •aubacriba'
    {
      aubacriba(gat_...u O,  $_Gn[  'id''J 1,
      diaplay~it... ('Subllcribad.Liata•,  gat_aubacribad_liatalgat_...1111,
                    'iDforaatiCll1',   '•bow-archiva•,   'UD8Ubacriba•I,
      break,
    }
    ca••  •uaaubacriba'
    {
      uaaubacribaCgat_...uO,  $_Gft['id'I  1,
      diaplay_it... l' Subacribad Liata', gat_aubacribad_lbta lgat_...u O 1,
            'inforaation•,   '•how-archiva•,   'unaubacriba'li
      break,

    cu-'!
    caaa  •ataow-111-liata'
    {
      di       it... 1'Subacribad Liata', gat_aubac:ribad_Uata Cgat_...11 C 11,
                    •in  oraat  oo•,  •   -•     ve ,                  ,

    }
    caaa  •cllange-paaaword'  ,
    {
      diaplay_paaaword_foraCl1
      break,
    }
    caaa  •atore-change-paa~'  ,
    {
      iflchange_paaaword(gat_...1111,  $_P08T1'old_paaawd'I,
        $_108T[•naw_paaawd•J,  $_108T['naw_paallwd2'JII'

        echo  •cp>Olt1  Paaaword changed.c/p>
          cbr  />cbr  />cbr  />cbr  />cbr  />cbr  />'1
      }
      alaa
      {
        echo  •cp>Sorry,  your paa~     could not ba chaDgad.c/p>' I
        diaplay_paaaword_foraCI;
      }
      braak!


}
// Laa aiguimtaa acciona• a6lo laa puada realhar  al uauario aclaiDiatrativo
ifCchack_adain_uaarCI~
(
```



```
      diaplay_..11_fora(get_...11c11,
      bnet,
    }
    caaa 'craata•liat•  ,
    {
      ~aplay_liat_fora(gat_ ...111111
      braat1
    }
    caaa 'atore-liat•  ,
    (
      iflatore_liatC$_..1s1011(adain_uaar•J,   ._1'OIITII
      (
        acbo  •cp>... liat_addadc/p>ebr  />'1
        diaplay_itaal!C'All Liau ', get_au_uau    0 , 'infonation  ',
                      'ahow-archiva','I;

      alaa
        adlo  'cJl>Liat  could not. ba atorad, plaaaa try '
            .•again.c/p>cbr  />cbr  />cbr  />cbr  />cbr  />l

      break,

    e...   '•md'
    (
      [redacted]   SUSIQ11[•adaiD  uaar•J 1 /
    }
    caaa 'viaw ·•il'
    (
      diaplay_it ... C'Unaant  11a11•,  gat_uaaaDt_..ilCget_ . . 11111,
                    'praviaw-htlll',   •praviaw'taxt•,   'aallll' ||
      break;


;..'...'...''..''..'.......''.'''.'...'...'...'..'.'..'..'.'''...''''.
' sacci6D ,,  — trar pia da pgina.
..'''.....'.........'.....'.''..'..'''.''.'.'...'.''''.'''.'''. ;

  do_htal_footarCI,
?>
```

Comprobad que en el listado se han separado claramente los cuatro segmentos del código. En la fase de preprocesamiento, establecemos la sesión *y* procesamos todas las acciones que se deban realizar antes de enviar los encabezados. En este caso, se incluye el inicio *y* el cierre de la sesión.

En la fase de encabezados, definimos los botones de menli que va el usuario *y* mostralllOSlos correspondientes encabezados por medio de la función do_html_header() de output_fna.php.Esta función simplemente muestra la barra de

encabezados y los menús, por lo que no la describiremos con detalle. En la sección principal de la secuencia de comandos, respondemos a la acción seleccionada por el usuario. Estas acciones se dividen en tres subconjuntos: acciones que se pueden realizar sin estar conectado, acciones que pueden realizar los usuarios convencionales y acciones que pueden realizar los administradores. Por medio de las funciones ckech_logged_in () y check_admin_user () comprobamos si se permite el acceso a los dos últimos conjuntos de acciones. Estas funciones se encuentran en la biblioteca de funciones user auth. fns.

El código de las mismas, y también de la función check_normal_user(), se muestra en el listado 30.3.

Uabldo 30.3. Funciones de user_auth_fns.php. Comprueban si el usuario ha iniciado sesión o no y a qué nivel.

```
funccion cbeck_normal_uaer()
// COIIPru•be sI hay alguien coneccado y, en caso concrario, nocifiqueeelo
(
   if (i•••C($_SESSIOH('normal_u••r')))
      recurn crue,
   el••
      recum false;


f ~ cion cbeck_admin uHr ()
// compruebe .r hay alguien coneccado y, en caso concrario¡ nocifiqueeelo
¦~
   if (i•••C($~SESSIOH('admin_u ••r•J)l
      recurn crue I
   el••.
      recurn false;


funccion check_logged_in(l
¡'
   recurn ( check_noraal_Ü•er(l 11 check_admin_user(l 11
```

Como puede comprobar, estas funciones utilizan las variables de sesión normal_user y admin_user para comprobar si el usuario se ha conectado o no. Veremos la configuración de estas variables más adelante.

En la sección final de la secuencia de comandos, enviamos un pie de página HTML con ayuda de la función do_html_footer() de output_fns.php. A continuación, en la tabla 30.2, se enumeran brevemente las posibles acciones del sistema.

En la tabla falta una opción como store-mail, es decir, una acción que cargue los bole~ informativos introducidos a través decreate-mail por los administradores. Dehecho, esta funcionalidad conaeta se encuentra en un archivo diferente, upload. php. La incluimos en un archivo diferente ya que resulta más sencillo para los programadores realizar el seguimiento de los aspectos de seguridad.

Tabla 30.2. Posibles acciones en la aplicac:16n de geetl6n de listn de c:oneo.

ENUM('valor1','valor2',...)		65.535
log-in	Todos	~ 11 UlU8rio un famwrlo de Inicio de .....
log-ouc	Todos	Fin811D-Nli6n
aew-accounc	Todos	Cree __ cuenta PII,. un UlUllrlo
acoze-accounc	Todos	~delmIIN dec:uentas
show-all-lists	Todos	Muellrm una Nlla de ... Islas de COI'NO clilpollllllN
ebow-archive	Todos	Muestra llalelllIN lnfarmlIU-ardlivados de_ .... __
informecion	Todos	MNstnl ilbmec:i6,, b6sic:8 ICllre una liste
ImageDestroy ($im);	Usuerloaconecladm	Muestni la conftgurac:i6n de la cuente del usuario
ahow-oeher-lieca	u-riosconectadOS	MullllnI lasllslasdeCOIT80alasque noftt6 lUSC:rilo el u-rlo
ahow·11Y·li•C•	u-riosconectados	Mueill'll las listas de correo a las que estt lUSC:rilo .. UlUllrio
subscribe	Usuarios conectados	SulC:ribeun UlUllrioa una lista detérminada
unsubscribe	Usuarios conectados	Anula•...afpc:i6ndeun.uartoa unalilta dlletn•iede
change·paHword	Usuarios conectados	MUNlra el formulario de cambio de conlra-IIIl'l8
ecore-change-paHword	u-riosconectaclos	AcluaUza la conll'llsalla del u—rio en la basededatos
creace-uil	AdminiStnldores	MuelClll un formulario para cargarbolellnn itdOlii ....
creace-lisc	Administradores	MUNlra un formulario pt1ra podercrurnue-va listas de c:onN
acore-Use	Administradores	~ detallesde... lislasdecorreoen la bnede datos
view-•il	Administradores	Muestra bolellne•inform.U- que se han ca,gaclOpen,que no se han IIIViado
	Administradores	Envfa boletlnN Informativos a los SUICriptoras

Analizaremos la implementación de estas acciones en los tres grupos indicados en la tabla 30.2, es detjr, acciones para usuarios no conectados, accio~es para usuarios conectados y acciones para administradores.

EL_HOLISTICO@HOTMAIL.COM - 2006

Implementar el inicio de sesión

Cuando un nuevo usuario accede a nuestro sitio, podemos esperar que realice tres acciones. Primero, que vea lo qtae ofrecémos; segundo, que se registre y, por último, qu~ se conecte. Veremos cada una de ellas de forma separada.

En la figura 30.4 se muestra la pantalla a la que acceden los usuarios cuando visitan el sitio por primera vez.

Figura 30.4. Al llegar, loa usuarios pueden crear una nueva cuenta, ver las llstas disponibles o simplemente conectarse.

A continuación veremos cómo se crea una nueva cuenta *y* cómo se realiza la conexión, *y* volveremos a la operación de ver detalles ~ las listas en un apartado posterior.

Crear una nueva cuenta

Si un usuario selecciona la opción de menú New Account, se activa la acción new-account que, a su vez, activa el siguiente código en index. php:

```
ca•e •new-account'  :
(
   // elimine laa variable• de aeai6n
   aea•ion_d••troy(|;
   di•play_account_for111||1
   break;
```


Este c6digo desconecta a un usuario si ya • encuentra conectado y muestra el formulario de detalles de cuenta (véase la figara 30.5)•

Figura 30.5.8 formularlo para ciear nuevas cuentas pennlte a los u~ Inlroducir 1111 dalol,

Este formulario se genera por medio de la fundóndisplay _account_form(l de la biblioteca output_fns. php. Esta función se utiliza tanto con esta acción como con account - set t inga para mostrar un formulario que permita al usuario configurar una cuenta. Si la función se invoca desde la acciónaccount-aettings, el formulario se completa con los datos de cuenta existentes del usuario.

En este caso el formulario está ~n blanco, listo para ~troducir ~uevos detalles de cuentas. Como esta función únicamente representa HTML, no la analizaremos aquí.

El botón de envio de este formulario invoca la acción store-account, cuyo código mostramos a continuación:

```
ca•• •atore-account• ,
(
   if ‹•tore_account($_SBSSION('nor91111_u••r'I,
        $_SBSSION1'adllln_uaer•J,  $_POST))
      $action • ••|
   if(!cbeck_logged_in111
      di•play_lo.gin_for111($action11
   break;
```

La función atore account () escribe los detalles de la cuenta en la base de datos. Su código se recoge en el listado 30.4.

EL_HOLISTICO@HOTMAIL.COM - 2006

Llatado 30.A. Funci6n ltar'e_accaunt() de mlm_fns.php. Eu func:i6n 8l'l8de un nuevo usuario o akMcena los demllH modiftcadoa de un usuarioya existente en la base de datos.

```
:unctiOII •tore_account($normal_Ul1er, $adain_1111er, $detail•I

   if(!fillecl _out($detailel)
   {
      ecbo 'All field9 au•t be filled in. Try again.cbr />cbr />';
      return fal••;

   •]••
   {
      ifC•ubllcriber_exiata(Sdetaila[•  ...11•111
      {
         //cO!ipruebe que •• ban conectado coiio el uauario que tratan de cambiar
         if(get_...ilCI• •$detaila[•...11•11
         {
          $query • 'update aubllcriber• ••t realname • '($detail•C'realna-'I}',
                                          aiiMtyp• • '($detailaC'llimetype'J}'
             wbere ...11 • '{$dataila[• ...11•1}'I
            if($conn•db_connect(II
            {
               if  C$conn ·:,query($queryl!
                  return trueI
               •]••
                  return falaeI
            }
            •]••
```

```
ecbo •c:oalll –t •tore cbange9.cbr /><br />cbr /><K />cbr />
cli,r />':
return falM1--.- - --  -- --- . . . .
```

```
         }
         .1 ..
         {
            ecbo 'cp>SOrry, tbat • —il addn•• i• already regiatered bere. c/p>';
                 'cp>You will need to login witb tbat addr••• to cbange '
                 ' ita aettinga.c/p>';
            return fal•• 1

      }
      •]•• // snaeva cuenta
      {
         $query • •inaert into aubacribera
                  valU1la ('{$detail•l'• •il'I}',
                           '($dataila[•realn&M')}',
                           •($detaila[•mi..type'J}•,
                           abal('{$dataila[•new_paaaword')}'I,
                           OI•;
         if C$c-·db _connectC11
         {
            if  CS– ·>query(Squeryll
               return true;
            •]••
```



```
            return falae;

         el••
         {
            ecbo 'Could not atore new account. cbr />cbr />cbr />><br />><br />
            cbr />';
            return falae;
```

En primer lugar, la función comprueba que el usuario ha introducido los datos requeridos. Si son correctos, la función crea un nuevo usuario o actualiza los detalles de la cuenta si el usuario ya existe. Un usuario sólo puede actualizar los detalles de la cuenta del usuario bajo el que se ha conectado.

Para realizar la comprobación se utiliza la función get_mail O, que recupera la dirección de correo electrónico del usuario conectado actualmente. Volveremos a este aspecto más adelante, ya que utiliza variables de sesión que se configuran cuando se conecta el usuario,

Iniciar sesión

Si un usuario completa el formulario de inicio de sesión que vimos en la figura 30,4 y püa el botén Leg In, ~r.ará la secuenda ele ,omandCIII index php COA lu variables email y paeaword configuradas. De esta forma se.activa el código de conexión, que se corresponde a la fase de preprocesamiento de la secuencia de comandos y que mostramos a continuación:

```
// u neceaario proceaar ante• que -da la• aolicitudea de con-i6n y daeconexi6n
if ($_POST'('—il')66$_POST('paH-rd') I
{
   $login• loginC$_POST('...il'I. $_POST('paaaword•)I;

   if($login • • 'aclmin'I
   {
      $atatUII •• '<p><b>'.gat_real_Dal1e($_POST('—il')1.'c/b>  logged in•.
                 ' aucceaafully aa cb>Admilliatratorc/b>c/p>
                 'cbr />cbr />><br />cbr />cbr />';
      $_SBSSIOll[•admin_uaer•J• $_POST('...il');
   }
   •l•• if($login ••'normal')
   {
      $atatUII •• •cp>cb>•.gat_real_na..($_POST('—il')l.•c/b>  loggecl in•
                .' aucceaafully.c/p>cbr />cbr />°;
      $_SBSSIOll('normal_uaer'J • S_POST[' ...u•J;

   elH
   {
      $atatU11 •• °<P>iSorry,,,.. coald not log you in witb tbat
                 ... 11 acldre•• &1111 paaaword.c/p>cbr />°;
```

```
1f($act1on  • • 'log·out'I
(
   un•et($actionl!
   un••tl$_SBSSIO1ll!
   ••••ion_d••troy[!;
```

Como puede comprobar, primero intentamos conectarlo con la función login ( ) de la biblioteca user_auth_fns. php. Es ligeramente distinta a las funciones de inicio de sesión que hemos utilizado anteriormente, por lo que la analizaremos con m6s detalle. En el listado 30.5 puede ver el código correspondiente a esta función.

Usudo 30.5. Funcl6n login() de user_auth_fns.php. Comp,obaci6n de los detalles de conexión de un usuario.

```
function loginl$• ..il,  $pa••word!
// CQ11Pru•ba •l n0llb~e da ueuario y la contra..lla en la baee de dato•
// ai coinciden, devuelva el tipo de conex16n
//  en caeo contrario, devuelva fal•e
|
   // contcteae a la ba•e de datoa
   $conn • db_connect O;
   1f  (!$COM)
      return  o,
-------- - — —--;:""!$qu¡;¡¡.,ery¡;;;-;;',-,oo¡¡¡e.-i1l-ic!!l1t""ll1[1Dli111t'b..-r,f:t""0ll..--.nsublhiaR'(C'"ni!..beorza-------
                                 where email ·'$email•
  ~                              and paeeword • paaeword!$paeeword'I";
   $r•eult • $conn·>query($queryl;
   1f  !!$rHult!
      return falae;

   if  l$r••ult·>n1111_rowecll
      r•tilrn  falee,

   $row • $reeult·>fetcb_array();                    ····

   if($row(OJ • • l)
      r•turn  ‹adain›
   •lee
      r•turn  ‹normal›
```

Anteriormente con las funciones de conexión devolvíamos true si la conexión era satisfactoria y false si no lo era. En este caso, devolvemos false si la conexión no ha sido correcta pero, en caso afirmativo, devolvemos el tipo de usuario, 'admin' o 'normal'. Comprobamos el tipo de usuario recuperando el valor almacenado en la columna admin de la tabla de suscriptores, para una combinación concreta de direcciones de correo electrónico y contraseñas. Si no se recupera ningún resultado, devolvemos false. Si el usuario es un administrador, este valor ser6 1 (true) y devolvemos 'admin'. En caso contrario, devolvemos •normal•.

Volvamos a la lfnea de ejecuc:ión principal. Registramos una variable de sesión para realizar el seguimiento de riuestn;,usuario.

Sed adllin user si se trata de un administrador o normal uaer si es un usuario convencional. Sea cual sea la variable, contendd la direcci6n de correo electrónico del usuario. Para simplificar la comprobac:iónde la misma. utilizamos la función get_mail (› que mencionamos anteriormente. Esta función se describe en el listado 30.6.

Llatlldo 30.1. Funcl6n get_emell() de user_aulh_fnL.php. Devuelve una dirección de correo electn)ilico del usuario conectado.

```
function get_e..1111
(
   if  lieeet($  sasSI011('normel  ueer'JI
     ratum $-SDSI011('normel-uer'JI
   if  liHet 1$-HSSI011('adain  ~ r'111
    return $_SBSSI011('adllin_uier•JI

   return falee;
```

De nuevo en el programa principal, informamos al usuario si esbl conectado o no, y a qué nivel.

El resultado de un intento de conexión se muestra en la figura 30.6.

Figura 30.1. El sl1tema Inform• ll usu•rio de que la conexi6n ha lido correen.

Una vez conectado el usuario, podemos paur a las funciones de usuario.

EL_HOLISTICO@HOTMAIL.COM - 2006

Implementar funciones de usuario

Oespuá de conectarse, los usuarios pueden realizar cinco operaciones:

- Ver las listas disponibles a las que se pueden suscribir
- Suscribirse y anular la suscripción a una lista
- Cambiar la configuración de sus cuentas
- Cambiar sus contrasei\as
- Desconectarse

En la figura 30.6 hemos visto la mayoría de estas opciones. A continuación analizaremos la implementación de cada una de ellas.

Ver las listas

Implementaremos una serie de opciones para poder ver las listas disponibles y los detalles de las mismas. En la figura 30.6 puede ver dos de estas opciones: Show My Usts, que recupera las listas a las que está suscrito el usuario y Show Other Lists, que recupera las listas a las que no est! suscrito.

Si se fija de nuevo en la figura 30.4, comprobará que existe otra opción, Show AH Lista;:que recupera todas las listas de correo disponibles en el sistema. Para que el sistema sea realmente escalable, será necesario añadir funcionalidad de paginación (para mostrar, por ejemplo, 10 resultados por página)... N.~ lo he~~ hecho por motivos de brevedad.

Estas tres opciones de menú activan las acciones show-my-lists, show-other-lists y show-all-lists respectivamente. Co¡no habrá apreciado, las tres acciones tienen un ñmcionamiento similar. A continuación se incluye el código de las mismas:

```
caaa •abov-all-liata•  ,
{
   diaplay_it...l'All Liata•, gae_all_liatal),  •information•,
                 'a110w-arcbiv.•.••)1
   break;


ca.. 'allov-otbar-liata•  ,
{
   diaplay_:ita-1•vnaubac:ribad r.iata•;
                 gat_unmub.c:ribed_li•ta(gat_a..illll,   •info1n1&tion•,
                 'abolr-arcbive', 'aub•criba'l1
   br..it;

e••••:
```



```
caaa •abov-my-liata•  ,
{
   diaplay_it...('Subacribad Liata•, gat_aubacribad_liata(gat_e..illll,
                 'information•, 'abov-arcbive•, 'unallbacriba'li
   break;
```

Como ve, las tres acciones invocan la función display_i tem( ) de la biblioteca ouput_fna. php, pero cada una lo hace con diferentes parámetros. Tambiin utili- ~ la función get _mai l (l. que hemos mencionado anteriormente para obtener la dirección de correo electrórucocorrespondiente a cada usuario.

En la figura 30.7 puede ver lo que hace esta función.

Figura 31.7. La función dlsplay_items() se ha utilizado pera mostrar una lista de las listas a las que no está suscrito el usuario.

Veamos a continuación el código de la función display i tems () que se incluye en el listado 30.7.

U.lado 31.7. Función dlsplay_ltems de output_fns.php. Esta función se utiliza pera mostrar una lista de elementos C:0ll sus c:orrespondlentas acciones.

```
function diaplay_it...1$title, Sliac, Saction1•'•  Saction2•'',
$acti—1•' 'I
(
   global $tabla vidtb;
   eclo •ctabla ;idtb•"$table vidtb" callapacing•"0" cellpad4ing••0•
         border.•o'~';

   // cuente al D1l-ro da acci-
```

```
$actiona • (($actiaal1•'')   + ($acti0ll21•''1   + ($action1l•''1l;
echo 'ctr ..
      cth c:olapan•"'·   (1+$actional  .•• bgcolor•"ISa,,~ ·..·.  Stitle .•c/th..
      c/tr..';

/I cuente el na.ero ele arUCllloa
Sit... • aiseof($liatl1

if($1te ..  • • O)
  echo 'ctr..
        ctd c:olapan-••.(1+$actioaa).••   align.•eenter•..
        II0 It—   to Diaplayc/tlb
        c/tr>';
elae
(
   //  impriM   toda•   laa   Ulaa
   for($i • 0,  $1t...,  $1++1
   (
      if!$il21   //  loa colorea ele fondo ae alternan
         $bgcolor • "'lffffff'"1
      elae
         $bgcolor • ••1ccccff'"1
      echo •ctr..
         ctd bgc:olor•$bgcolor
         width•\••.(Stable  width -  ($actiona•149)).••.,';
      echo $1iat($1l  [111  -
      if ($1iat  [$1) [21 1
         echo•  -  .•.$1lit!till21,
      echo 'c/td ..!1

      // cree botonea para una IIAximo ele trea accione• por linea
      for($j  • 1;  $jc•l1  $j++I
      (
         $var • 'action!.$j1
         if($$var)
         (
            echo •ctd bgcolor•\"Sbgcolor\• width•\"149\•..',
            // loa botonea view/preview aon un c~o eapecial ya que eatAn
            // vinculadoa a un archivo                    '
            if($$var  • •  'preview-htlll'1l$$var  • •  •view-htlll'II
               $$var  • •  •preview-text'!l$Svar   • •  'view-text•)
               diaplay_preview_buttaa1$liat[Sil  [JI,  $11at[$1) [0),  $$varl1
            elae
               dlaplay_buttaa( $Svar,  ••id•'  . Sliat[Si)  [0)  );
            echo 'c/td..'!

      )
      echo •c/tr..\n•1

   echo •e/table..',
```

Esta fundón repiesenta una tabla de elementos, cada uno con hasta tres botones de acdón asociados. La función espera dnco par6metros que, en orden, son los siguientes:

- $title es el lftulo que aparece en la parle superior de la tabla. En el caso ilustrado en la figura 30.7, pasamos el lftulo UlllUllecribed Lista, como indicamos en el fragmento de código desaito anterionnenle para la acción "Show Other Lisis••
- $Úat es una matriz de elementos que se muestra en cada fila de la tabla. En este caso, ee trata de una matriz de laa Datas a las qae el usuario no est4 suscrito actualmente.

  Generamos la matriz (en este caso) en la funcidn get_unaubacribed_lista (), que veremos en breve. Se trata de una matriz multidimensional en la que cada fila puede conlener hasta cuatro fragmento& de datos sobre cada fila. En orden. son los siguientes:

  - $1iat (n) (O) contiene el Id. del elemento, que normalmenle es un nwnero de fila. Proporciona a los botones de acción el Id. de la fila en la que ac:hían. Ennuestro caso, utilizuemos el Id. de la base de datos, como veremos en breve.
  - $liat (nl (1) contiene el nombre del elemento. Ser6 el texto queaparezca para un determinado elemento. Por ejemplo, en el caso de la figura 30.8, el nombie del elemento en la primera fila de la tabla es ————.
  - $1iat (n) (2) y $li at (nJ (3) son opcionales. Los uti1'áremos para indicar que hay mú informadón. Se corresponden. respectivamente, al ~~~to~,~~~l~d~.dde~~k~~~~~~~~~aKd~6~DWIMdliidd~~llJ~~~~~LIUR~~~-~~p~~~en~~e~l~~~~ que se utilizan esto& dos par6metros cuando analicemos la acción View Mail, en un apartado posterior.
- El tercer, cuarto y quinto pamnetros de la fund6n se utilizan para pasar tres acciones que se mostrar6n en botones correspondientes a cada elemento. En la figura 30.7, se trata de los tres botones de opc:i6n representados como Infonnatlon, ~how Archive y Subacrlbe.

Obtenemos esto& tres botones para la página Show All Lisis al pasar los nombres de acción, inforaation, abow-arcbive y aubacribe. Al utilizar la función display_button (), estas accionesse han convertido en botones con dichas palabro sobre los mismos, y ee les ha Dignado la correspondienteacdón.

Todu laa acciones Show invocan la función display_itemde forma düerente, como puede comprobar si observa sus acciones. Ademú de tener títulos y botones de acción diferentes, cada una de ellaa utiliza una función distinta para generar la matriz de elementos que se van a representar. Show All Liats utiliza la función get_all_lists(), Show Other Lisis utiliza la función get_unaubacribed_liata () y Show My Lista, la función get_aubacribed_liata (). El funcionamiento de lu tres fundones es similar y todas provienen de la biblioteca mlm_fna .pbp.

Describiiemos get_unsubacribed_liata (), ya que es el ejemplo que hemos seguido hasta ahora. Enel listado 30.8 se muestra el código de esa fundón.

EL_HOLISTICO@HOTMAIL.COM - 2006

Uatado 30.8. Funcl6n get_unsubscribed_llsts() de mlm_fns.php. Genereción de una malriZ de listas de correo a las que el usuario no esd suscrito.

```
//Obtenga la• llataa a laa que no eat6 auacrlto ••t• uauario
functioa get_uaaubacribed_liata($eeaill
(
   Sliat • arrayll,

   Squery • •aelect llata.liatid,   liataa..,  ...11
         fr-  liata  left
         join aub_liata
            oa liata.liatid  • aub liata.liatid   and
               eeail•'$email'  vhere emall ia NULL order by liataa..•;
   if 1$coaa•db_coanect 111
   (
      Sreault • $coaa->query($queryl:
      it(l$reaultl
      (
         echo •cp>Unable to get liat trom databaae.c/p>';
         retum  tal•••
      |
      $aua • sreault->aua_rowa;
      tor($1 • 0, $1c$nua; $1++1
      (
         $row • $reault->fetch_arrayll:
         array_puahl$liat,  arrayl$row(OJ,  $row(llll1


   retura $llat1
```

Como puede comprobar, esta función requiere que se pase una dirección de correo electrónico. Debería ser la dirección de correo del suscriptor que estamos utilizando. La función get _suscribed _li sts () también requiere una dirección de correo como parámetro, pero get_all_lists () no, por razones evidentes.

Tras indicar la dirección de correo del usuario, nos conectamos a la base de datos y obtenemos todas las listas a las que el suscriptor no está suscrito. Como suele ser habitual con este tipo de consulta en MySQL, utilizamos LEPT JOIN para buscar elementos que no coincidan. Realizamos un bucle por el resultado y generamos la matriz fila a fila por medio de la función incorporada array _push ( ).

Una vez que sabemos cómo se genera esta lista, veremos los botones de acción asociados.

Ver información de listas

El botón lnfonnatian de la figura 30.7 desencadena la acdón infon:ation, que mostramos a continuación:

```
çaae 'iaforaatioA'  :
(
   diaplay_inforaaticn(S_GBT[' id•J 1:
```



```
break,
```

Para ver lo que hace la función display _information (), observe la figura 30.8.

FIpra 30.8. La función dlsplay_information() lllil8Stla un comaAlariuoln una lista da correo.

La función muestra información general sobre una determinada lista de correo •.. así como el número de suscriptores y el número de boletines informativos que se han enviado a dicha lista y que se encuentran disponibles en el archivo (que veremos en breve). El código de esta función se recoge en el listado 30.9.

Ustado 30.9. Función display_infonnatlon() de output_fns.php. Muestra inlonnación da la lista.

```
tunction diaplay_intormation($liatidl
(
   re(!$liatidl
      retum  talae,

   Siafo • load_liat_lafo($liatidl;

   it1$intol
   (
      echo 'ch2>'.pretty($1afo[liataamell.'c/h2>'1
      ecbo '•P>'.pretty($iato(blurbll'c/p>1
      ecbo 'cp,.llllalber of aubacribera,•  ' $lnfo(aubacriberal'•/P>:
      echo •cp,.lhlllber of eaaaage• in archive:'  . $info(arclaive)'c/p>;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

La funci6n di•play_information () utiliza otru dos funciones para <OIIH-guir sus propósitos Web: load_lht_info (l y pretty().

La fund6n load_limt_info() se encarga de recuperar datos de la bue de datos. La función prettf O simplemente aplica formato a los datos recuperados, elimina lu barras Invertidas, convierte nuevoe púrab en saltos de linea HTML.etc.

Veamos brevemente la func:i6n load_liat_info(), que se encuentra en la biblioleea de funciones mlm_fn• •php. Su c6digo 1e Incluye en el listado 30.10.

LIst8do ao.11. Funcl6n loecl_Nst_Info() de mlm_fns.php. Gener9Ci6n de .... matriz de Infonnecl6n de llst8.

```
function load_li•t_infoCSli•tidl
(
   if C!Slhtidl
      ntum fal••,

   ifCIC$ccmn-dlil_cODDectCIII
      return falaa,

   Squ•ry • •••l•ct li•tna.., blurb fro11 li•t• when li•tid • Sli•tid•;
   SrHult • $conn->queryl$query);
   if (! $rHultl
   (
      echo 'cannot retrieva thi• liat•;
      ncura false,
   }
   Sinfo • Sr••ult->fetch_a•soc,11

   Squery • ••elect countC•I fro• .ub_li•t• when li•tid • Sli•tid•,
   re•ult • Sconn->query(Squeryl;
   if ($reeultl
   (
      Srow • Sra•ult·>fetch_array(I,
      Sinfol'•ub•criber••J • $row[O11


   $que~• ••elect count(•I froa aail where li•tid • $li•tid
          and •tatu• • '111111''"1
   SrHult • Sconn->queryl$query),
   if!Sn•ultl
   (
      Srow • Sn•ult·>fetch_array(I;
      Sinfo[•archive•J • $row[0)1

   retun Sinfo;
```

Esta función e;ecuta tres consultu de bue de datos para obtener el nombre y el comentario de una lista de la tabla lhu, el nwnero de suscriptores de la tabla •ub_liata y el número de boletines informativos enviados de la tabla mail.

Ver archivos de listas

Adema de poder ver el c:omenlario de la lista, los usuarios pueden CG111probar todo el correo que se ha enviado a una determinada lista si pulAn el bolml Show Ali:ldve: Sé activ.u' la acción ahow-arcbi,re,que desencadena el liguimle e6digo:

```
caM '•llov-archive' :
(
   dbplay_it-i•Archive Por ',!19t~U•t__ l$_Gft|'id'11,
        gat_archive($_GIT('id')I, •viaw-htal', •viaw-text', ''11
   break;
```

De nuevo, esta función utiliza la función dbplay_iteu (› para enumerar los distintos elementos de correo que se han enviado a la lista. Estos elementos se recuperan por medio de la función get_archive () de mla_fnm •php, cuyo c6dl-go reproducimos en el listado 30.11.

Ustado 31.11. Func16n get_an:hlve() de mlrn_fns.php. Generaci6n de una matriz de balellnea lnfonMlivaaen:hlvedoa pera una determnacle llllll.

```
function get_archiveCSli•tidl
(
   //devuelve una aatrh del correo archivado para Hta Una
   //la •tris tiene fila• c- Cid de correo, uunto|

   $11-t • arrayll;
   $11•tname • get_liat_n ... l$li•tidl1

   Squery • "••lect aailid, 9ubject, liatid
            from aaU
            when li•tid. Sli•tid and •tatu• • 'SIIIT' arder by MDt°1

   if(Sconn•dlil_cc:mnec:tll)
   (
      Sneult • Sconn-..queryCSquery)1
      if (!$rHult)
      (
         echo •cp>Unabl• to get U•t froa databa•• • $query.c/p>°I
         ntum falM;
      }
      Sn1111 • Sreeult->n1111_row91

      forCSi • o, $ic$nua1 Si++)
      (
         Srow • Sreeult·>fetch_arrayCl,
         Sarr_row • array(Srow[OI, Srow(11,
              ·Slin-. SUnidl I
         array__puah!$li•t, Srow1,

      }
   return $lbt1
```

Como en el caso anterior, esta función recupera la información necesaria de la base de datos, los detalles de correo que se han enviado, y genera la correspondiente matriz que se pasará a la función display_items.

Realizar y anular suscripciones

l;n la lista de listas de correo de la figura 30.7, cada lista dispone de un botón que permite a los usuarios suscribirse a la misma. Del mismo modo, si los usuarios recurren a la opción Show My Lists para ver las listas a las que ya están suscritos, verán un botón Unsubec:ribe junto a cada una de las listas. •

Estos botones activan las acciones subscribe y unsubscribe, que desencadenan los dos siguientes fragmentos de código, respectivamente:

```
caae 'aubacribe' :
{
   aubacribe(get_e ..il(I, S_GBT('id'II;
   diaplay_it...('Subacribed Liata', get_aubacribed_liata(get_e ..il(II,
         'information•, '•how-archive', •unaubecribe'I;
   break;
}
caae •unaubacribe' :
{
 .. unaubacribe(get_e • UO, s_GBT('id'll;
 'diaplay_ite ..c•subacribed Liata', get_aubacribed_liata(get_e • il(II,
 ..  'in!!LrJYtiq_~·, ·~hqw-..rc_l;lj_vll!', •1,m•~ci;1,'a'l..l. ·--
   break;
```

En cada caso, invocamos. una función (subscribe () 0. unsubscribe ()) y, tras ello, volvemos a mostrar una lista de listas de correo a las que el usuario se ha suscrito por medio de la función display_ i tema ( ) .

Las funciones subscribe () y unsubscribe (l se muestran en el listado 30.12.

LIstado.30.12. Funciones subscribe() y unsubscribe() de mlm_fns.php. Estas funciones alladen o borran suscripciones de un usuario.

```
function aubecribe($...il, $liatidl
{
   if(l$eMillll$liatidfllliat_exiata($liatidllll•ubacriber_mciata($e    ..illl
      return fal_,

   //ai ya eatA au.ecrito, aalga
   if(aubacribed($e..il, $liatidll
      return fal-;

   if (l$conn-db c:annect()))
      retu:na fal•e,

   $~ry • "in-rt into aub_li•t• valuea ('$...il•, $liatid)•;

   Srenlt • s-·:,,query(Squeryl,
```



```
      return $reault;


function unaubecribe(S...il, Sliatidl
{
   if(l$ ...illf1$liatidl
    ~ r•urn falae,

   if(!($conn•db_connectllll
      return falae,

   $query • "dlllete froa aub_liata wbere ...u•  •s...ll• and liatid. $liatid•l

   $reault • $conn-..queryl$query);
   return $reault;
```

La función subscribe () añacle una fila a la tabla sub_lists que se corresponde a la suscripción; la función unsubscribe ( ) elimina esta fila.

Modificar la configuración de una cuenta

Al pulsar el botón Aa:oant Settinp se activa la acdón account-settinga, cuyo código mostramos a continuación:

```
- —— -•accowat aettinga• '
 {
   diaplay_account_fora(gec_...ul1,
      get_real_na..(get_e..il(ll, get_ai..type(get_...illllll
   break;
```

Como puede comprobar, volvemos a utilizar la función display _account_ form () que utilizamos para crear la cuenta inicialmente. Sin embargo, en esta ocasión pasamos los detalles actuales del usuario, que se representarán en el formulario para facilitar la modificación. Cuando el usuario pulsa sobre el botón de envío del formulario, se activa la acción store-account como mencionamos previamente.

Modificar contraseñas

Al pulsar el botón Chmge Puswoni se activa la acdón change-password, que desencadena el siguiente código:

```
caae •cbange-paaaword• :
(
   diaplay_paaavord_foral);
   break;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

La función display_password_form() (de la biblioteca de fun(iones output_fna. php) simplemente muestra un formulario en el que el usuario puede cambiar su contrasella. Este formulario se il1!5tra en la figura 30.9.

Order processed at 13:54, 26th March

Your order is as follows:

En la secuencia de comandos, utilizamos echo para imprimir el valor introducido por el usuario en cada uno de los campos de formulario, seguido por texto explicativo. Si examina de cerca las instrucciones echo, verá que el nombre de la variable y que el texto que sigue están separados por un punto (.), como se puede ver a continuación:

FlgUl'II 30.9. La funci6n dlsplay_pessword() permlte aloa üsüanc:,scambiai' sus contrasellas.

C~nd~ el usuario p~ el bo~n a..nge Puswon:l, que se encuentra en la pa~ mfer1or.de! formulano, se activa la acción store-cbange-password. cuyo c6digo es el sagu1ente:

```
ca.. •atora-changa-paaavord•
(
   if|changa_paaaword|get_e ..11(1, $_POST('old_pa1awd•J,
       $_POST[ •new_¡,aaawd• J, $_POST[ •new_¡,aaawd2•J) l

   echo 'cp>0lt: Paaavord changad.c/p>cbr /><br />cbr /><br />cbr />cbr />'.
   }
   elae
   (
      echo •cp>Sorry, your paaaword could not be changad.c/p>';
      d1aplay_paaaword_for11(I:

   break:
```

Co~o puede comprobar, este código intenta cambiar la contrasefta por medio de la función change__paasword ( ) e indica al usuario si la operación ha sido satisfact~ria o no. La función ~hange_paHword ◁) se encuentra en la biblioteca de funciones user_auth_fns. pbp y su código se reproduce en el listado 30.13.

Uatado 30.13. Func:16n change_paaword() de user_autlt_fnl.pllp. Ell8 funcl6n v8llcla y .clu9llm la contruella de un usuario.

```
function cballga_paaavord($...il, Sold_paaavord, $aew_paaavord,
                                  ·~w_paaavord_confl
// cambie la contranlla dal correo alectr6nico/old_paa-rd por naw_paHvord
// davual,ra trae o faln
(
   J/ ai la contraala antigua ea correcta
   // callbia n contraaala por naw_paaavord y devllelTa trua
   // en caao contrario, dllYUalTa falaa
   if llogin1$-il, $01d_paaavordl1
   (
      if1$1119w_paaavozd- •Snaw_paaavord_confl
      (
         if Cl 1$c- • db connact O II
            raturn falaa;-
         Sqaery • 'updata aubacriban
                  aat paaavord • paaaword('$naw_paaavord'I
                  wller• ... u • '$... 11 ' •;
         $raault • $conn->quaryl$queryl,
         ratum $ranit:

      alae
         echo '<P> Your paaavorda clo not match.c/p>:

   al••
      echo •cp> Your old paaavord ia incorract.c/p>'1

   return falaa, // la contranlla antigua era incorrecta
```

Esta función es similar a otras funciones de configuración y modificaáón de contraseftas que ya hemos descrito. Compara las dos nuevas contrasellas introducidas por el usuario para garantizar que son idénticas y, en caso de que lo sean, intenta actualizar la contraseAa del usuario en la base de datos.

Cerrar sesión

Cuando un usuario pulsa el botón Log Out. se desencadena la acdón log-out. El código ejecutado por esta acción en la secuencia de comandos prindpal se encuentra de hecho en la sección de preprocesamiento, como ie indica a continuación:

```
if($action • • 'log-out'I
(
   un..t ($actionl :
   unaet($ SISSI0111
   aeeaion:deatroyl1:
```

Este fragmento de código elimina las variables de sesión y destruye la sesión. También anula la variable act ion,, lo que significa que introducimos la instrucción de caso prindpal sin una acción, con lo que se desencadena el siguiente código:

EL_HOLISTICO@HOTMAIL.COM - 2006

```
detault:
{
   if(lcheck_logged_1nlll
      d1splay_login_fo:na($action1;
   break;
```

De esta forma se podr, conectar otro usuario o el usuario podr.i conectarse como alguien diferente.

Implementar funciones administrativas

Si un usuario se conecta como administrador, tendr, acceso a opciones de menú adicionales, como puede comprobar en la figura 30.10.

Figura 30.10. El rMnú de administración permite crear y mantener llstas de correo.

Las opciones adidonales son Create List (para crear una nueva lista de correo), Create Mail (para crear un nuevo boletín informativo) y View Mail (para ver y enviar boletines informativos creados pero todavía sin enviar). Veremos cada una de estas opciones individualmente.

Crear una nueva lista

Si el adrnhústndor opta por definir una nueva lista y puJsa el botón Cleale Liat, activar, la acción **create-list**, que est.l asociada al siguiente código:


```
case 'create-list'   :
{
   display_liat_fona(get_email(II;
   break;
```

La función **display_list_form** (l muestra un formulario que permite al administrador introducir los detalles de la nueva lista. Se encuentra en la biblioteca **output_fns**.php y simplemente representa HTML por lo que no la describiremos. El resultado de esta función se muestra en la figura 30.11.

Figura 30.11. La opción Create Llst requiere que el administrador introduzca un nombre y una descripción (o comentario) sobre la nueva lista.

Cuando el administrador pulsa el botón Save Liat. se activa la acción **store-1 is t**, que desencadena el siguiente código en index. php:

```
ca•• '•cor•-11•t'   :
(
   if(•core_li•t(S_SISSIOll['adllin_u•er•),    $_POST))
   {
      ec:llo 'cp>Rew list addedc/p>cbr />';
      display_it...l'All Lists'.  get_all_liatall, 'infonaation•,
                    ••how-archive•,••);

   el••
      ecbo 'cp>Liat could not be •tored, ple••• try '
           'again.c/p>cbr  />cbr  />cbr />cbr  />cbr />';
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Como puede comprobar, el código intenta almacenar los detalles de la nuéva lista *y*, tras ello, muestra la nueva lista de listas. Los detalles de la lista se almacenan por medio de la función atore_liat (), c:uyo código reproducimos en el listado 30.14.

Uat.do 30.14. Función atore_HII() de mlm_fn1.php. Esta función eftede una nueva Hita de correo • la base de datos.

```
{unction •tor•_li•tl$admin_uaer,   $detail•]

   ifltfilled _out1$detail•l>
   {
      echo 'All  field8 llll•t be fllled  in.  Try again.cbr  />cbr  />',
      return fal•e;

   el••
   {
      1f(lcheck_aclmin~u•er($acllllin_u•er))
         return fal•e;  •
        11 ¿cdmo pudo invocar Hta  funci&l alguien no conectado caao adllini•trador?

      {f11 ($conn•clb_connect ()))

         return  fal•••

      $quezy  •  ••elect  count(•I
                 fr-  li•u
                wbere  li•tna ..  •  '",$detall•('name•].•••)
      $re•ult  •  $ccnn ·>query1$query|:
      $row  •  $re•ult·>fetcb  arrayll,
      ifl$rcw(OJ  > 0)
      {
         ecbo  'sorry,  tbere 1• already a li•t  witb  tbi•  na.... ,
         return fal•e;

      $query  •  'in•ert  into li•t•  value•  IIIULL,
                  '{$detail8(na11e•J)   ',                  ...
                  '{$detall•('blurb)>i•1
      $re•ult  •  $conn ·>query($query>,
      return  $re•ult;
```

Esta función realiza algunas comprobaciones de validación antes de escribir en la base de datos. Comprueba que se han introducido todos los detalles, que el usuario actual es un administrador *y* que el nombrede la lista es exc:lusivo. Si todo funciona correctamente,la lista se aftade a la tabla 1 ists de la base de datos.

Cargar un nuevo boletín informativo

Por último llegamos a la parte principal de la aplicación: la carga *y* el envío de boletines informativos a listas de correo.

Cuando un administradorpulsa el botmCnate MaU. se activa la ac:c:idn c:reate-mail, c:omo se indica a c:ontinuac:ión:

```
c..e  •create-..11'  :
{
   dieplay_..il_fora(get_e ..11());
   break;
```

El administrador verá el formularioque mostramos en la figura 30.12.

Figura 30.12. la opción Create Meil ofrece al administrador une Interfaz para cargar archivos de boletines Informativos.

Recuerde que para esta aplicación asumimos que el administrador ha creado un boletin informativo en formato HTML *y* en texto, *y* que cargará ambas versiones antes de enviarlo. Hemos optado por implementarlo de esta forma para que los administradores puedan utilizar su software preferido para crear los boletines informativos. De esta forma la aplicación resulta más accesible.

Comprobará que el formulario cuenta con una serie de campos que el administrador debe completar. En la parte superior se incluye un cuadro desplegable para seleccionar la lista de correo. El administradortambién tiene que indicar el asunto del boletín informativo (lo que se utilizará como linea de asunto del correo electrónico posterior).

Los campos restantes son campos de carga de archivos, c:omo indica el botón Browse... junto a cada uno de ellos. Para poder enviar un boletín informativo, el administrador debe incluir tanto la versión en texto como la versión HTML del mismo (aunque lo puede cambiar para ajustarlo a sus necesidades). También hay

EL_HOLISTICO@HOTMAIL.COM - 2006

w,a serie de campos de imagen opcionales en las que el administrador puede cargar todu las mgenes que haya incrustado en el HTML. Será necesario especificar y cargar individualmente cada uno de estos archivos.

El formulario es similar a cualquier formulario de carga de archivos convencional a excepción de que, en este caso, lo utilizamos para cargar varios archivas. Por ello habm que realizar algunas modificaciones en la sintaxis del formulario y en la forma de procesar los archivos cargados.

El código de la función display_mail_form() se recogeenellistado 30.15.

u.tado 30.15. Función cllsplay_mell_form() de output_fna.php. &qa fUncl6n muellra ei formularlo de car;a de archivos.

```
functioa di•play~mail_fora($email, $liat1d•Dl
(
   // muaetre un forl1111ar1o html para cargar aa ma.vo ..a.aje
   global Stable_vidth;
   $11•t • get_all liat•II;
   $liete • aizeof($liatl1
?>
   ctü>l• cellpa441ag•°4" cellapaciag••O• border••O•
      width•"c?pbp echo $table_v1dtb?>•>
   cfdrai eactype•'19Ultipart/foll:1a-clata•  act1oa•'apload.pbp•  Mt.hock 'po8it' >
   ctr>
   :' ctd bgcolor••tcccccc•>
      Liat,
—c:ltd>
      ctd bgcolor••tcccccc•>
         caelect aa  •11et•>
         --?pbp
         for($i • 0; $1dlbta;  $1++1
         (
            echo 'captioa value • '.$11at[$il  [DI;
            1f !$liatid- • Sli•t [$11 [DI l ecbo ' Hlected• l
            acbo '>'.$11at($il  [11 .•c/opt1oa>\n•;

         ?>
         c/aelect>
      c/td>
   c/tr>
   ctr>
      ctd bgcolor••lcccccc°>
         Subject,
      c/td>
      ctd bgcolor•°lcccccc•>
         ciaput typ..•t••t• aa.e••aubject• value••c?pbp echo Seubject?>°
          •iz• • ,o ></td>
   c/tr>
   ctr>ctd bgcolor~•lcccccc">
      T-t Veraioa.,
   c/td>ctd bgcolor••lcccccc•>
      cia.put type•f1le a.a...•uaerfile[OJ• •iza•  ,o>
   c/td>c/tr>
   ctr>ctd bgcolor•"lcccccc">
```



```
      11'1'NL Veraioa.,
   c/td>ctd bgcolor•"lcccccc">
      ciapat type•file Dame•'uaerfile[ll'  aize • ,o>
   c/td>e/tr>
   ctr>ctd bgcolor--lcccccc• colapaa•"2">Ieagae, (optioa.all

c?pbp
   $11mC_i1111g•• • 10;
   for(Si • 0; $ic10;  Si++!
   (
      echo •ctr>ctd bgcolor•'lcccccc'>Image •  ($1+11  .• c/td>';
      echo •ctd bgcolor•'lcccccc'>";
      echo •ciaput type•'f11•' 11&118•'uaerf1le[•.($1+2).•J•  elze•''°'></td>
      c/tr>•;


   ctr>ctd colapaa-•2• bgcolor•"lcccccc• aligae•ceater•>
   ciaput typ••"b1ddea.• na...',... imagee• value••c?pbp ecbo S..a_1eagea?>°>
   ciaput type••biddea• name•"liatid•  valu••°c?pbp echo $lietid?>°>
   c?pbp diaplay_form_buttoa(upload·f1lea•I;  ?>
   c/td>
   c/forai>
   c/tr>
   e/table>
c?pbp
)
```

Conviene destacar que los archivos que queremos cargar introduc:irm sus nomt;resen Unasene4eenfffi1D de ttpo file y queb nombreSeStuú, cou,pteudidos entre userfile (OJ y userfile [ni. En definitiva, tratamoe a estos campos del formulario de la misma forma que a casillas de verificación y les uignamos nombres utilizando una convenci6n de matrices.

Si quiere cargar varios archivos por medio de w,a secuencia de comandos PHP, tendrá que utilizar esta convención.

En la secuencia de comandos que procesa este íormulario, obtendremos tres Inatrices.*A* continuación analizaremos dicha secuencia de comandos.

Procesar la carga de varios archivos

Recordará que el c6digo de carga de archivos se encuentra en un archivo independiente. El listado completo de dicho archivo se muestra en el listado 30.16.

Umclo 30.11. uploacl.pttp. Esta uc:uencla de c:omendoS cerve todos los archivos necesarios para un boletln informallvo.

```
c?php
   // eata fUDC1oa.al1dad •• ia.cluye ea un archivo diatiato,  lo que no•
   // pewite ••r •• paranoico• al reepacto
   // •i algo ila —l, aaldreaoe

   $. x_•1se • 50000;
```

```
iDc:lucle ('iDClude_fa•.pbp'I;
••••ion_etartCI,

// 11610 lo• u-ri- ele ... iainraei6D pueclell cargar arclliWII
if(lcbecll:adllia_ueerlll
{
   ecbo 'y— do -t ••-to be autboriaed to ue thi• paga.',
   utt,


// crN 1- botoa•• ele la barra ele berraaieatu de dlliainraci6D
•buttOD8 • arrayCI,
$butt0118(01 • 'cbaat•-pa•8WOrd'l
Sbuttoae(ll • •create-li•t'l
$butt0118(21 • •creata-..11•,
Sbuttoaelll • 'view-..u•,
$buttoae(CI • '10,-out•,
Sbuttoae(SI • '•bow·all·liet••,
$buttOD8(IJ • 'ebow•ll)'•liet8J
•buttoae[7J • '•bow-otber-li•t•',

do_htal_header('Pyraaid·NLN - Upload Pile•');

4isplay_t-lbarl$buttonel;

// compruebe que la p&9iaa н iavoca - lo• dato• ~•earioe
ul1s rl ....[•unrfil•'ll'-'I IOI

111,:P08T('nb:lect•J ll1,_,on('Un•J I

ecbo !Probl_, You 414 —t fill —t tb8 fora fully. 'l'lle iaat•• are the
only optioaal fielu. lach _ ..,. n...s. a eubject, tut venioa
aad u arna. vereioa. ',
do_htal_f-ter(l1
uit,


,11et • $_P08T['liet·J,
S•ub:lect • $_PCST('•ub:lect'l I

1ff1($CODD-clb_COIIDect(I))
{
   ecbo 'cp..C-ld llot COIIDeCt to 6c/p>' ,
   do_htal_f-ter(l1
   exlt,


// dada loe detalle• del corno a la baH de datoe
Squery • •iuert i11to ➖U valuee (IIIIILL,
      •••• SB8SIOll['adaiD ueer•J.••,
      '".$-.bject.••,  -
      .•.• li-t.••,
      'STOll8D' , IIIIILL, IIIIILL)•,
$renl't • •coaa->query(Squery1;
if CI Sre•ult1
{
```

```
clo_hCll1_f_terCI,
eaIt,


//abtea,a •1 Id. IlylQL ui,..... a uta correo
...1114 • $coll1l•>il18ezt_id1

ifCI ...Uidl
{
   do_lltlll_f~erll/
   uit,


// ei lete - .. el pri..r -.aea:I• archivado, fallar& la creaei&a del
//directorio•• correc:to
• lllrdir(archive/•.,liet, 070011

// ei la creaci6D del directorio eepeclfico para ••t• correo falla,••
// UD probl-
if(lllrdir('archive/1.,liet.•/ ...ilid•, 070011
{
   clo_htal_f~erll1
   utc,


// proceN una iteraci6D·n la ➖tria de archivos cargado•
   = 0;
while ($ FILES['userfile']['name'][$i]&&
      S:.PILIS['UNrfile!][➖—·)[$i)l•'DODe')

   ecbo 'cp>UploadiDf '.$_PILB8('UHrfile•](•-·]( $i].' ' '.llf
   ecbo , PILSS(•unrfile'l [leiael)($iJ.• hytN.c/p>'l
   if ', PILIS('unrfile'll'eiae•J(Sil• •OI
   {
      ecbo 'hobl_, '.t_PIL18('UHrfile•J 1·—·J 1$11.
         ' ie ••IO leagtll' ,
      $1++;
      caatiaue,
   }

   (f ($_PIL18('UNrfile•](•eiae•J($1)>'9ax_eiae)

      ecbo 'Probl_, '., PILIS('ueerfUe•](•-·)(•11 ·' 1• -r ·,
         .$.ax_eiae.'-hytee•,
      $i++;
      continue;


   // deberl- camprobar que la iM,aD carpde н una 1-v-
   // ei 9etlaa,eeiael1 puede cleteraiaar •• t-'lo, probabl-te lo Ha.
   if($i>l ..1,acl..,..1ael$_PIL&l('ueerfil•'l('t11p_11111111•J(•1111
   {
      acilo 'Probl_, '.$ PILU('ueerfUe•l(•11111111•)($iJ.
         ' ie corrupt,-or DOC a tif, jpef or 11119'l
      $1++,
```

```
// fil• o (el ...n•aje de textol y fil• I (el ..nsaje htlll) •on ca•o•
// eqecialH
ifCSi• •0)
   Sde•tination • •arc!live/$li•t/$..ilid/text.txt•;
•l•• if($i •• 1)
   $de8tiilation • •arc:ive/Sli•t/S..ilid/index.htal•;
el••
(
   Sde•tination • •arc:ive/$li•t/$..ilid/•
         .S f'ILIS['uHr!ile'J ['UM'I [$il;
   $query .-•in•ert inc~ I.. ge• value• !$..ilid;
            ".$_FILIS('"J8erfile') ('na•'I [$i).••,
            ".S_FI:.!S('u•erfile') ['type•J [Sil .••J•;
   sr.-lt • Sco1111-:,,que:r:y(Squeryl;
I

//•i utilizaaoa PH~ ver•i6n >• 4.03

if !li•_uploaded_filelS_FILES['u•erfile'l ['tmp_na..•J ($1])1
(
   // po•ible ataque de carga de archivo• detectado
   echo 'somethinc¡ fuc:y happenin9 vith '
         .$_FILES('user!ile•J [•name•J.', not uploading.•;
   do_btml_footerll:
   exlt;


:aove_uploaded_filet$_fILES['u••rfile'll'tap_na•'I (Sil,
                        $deacinationl:

// •i •• la verai6n <• 4.02
copy CSu•erfile[Si), Sde•ti~ationl;

unlinlrl$u•erfile($i)I:
•!

$1++;


di•play_previev_buttonl$li•t, S.. ilid, •previev-html'I,
di•play_previev_buttonCSli•t, $..ilid, •previev-text'I;
diaplay_buttonC•aend', ••id•$..ilid"I;

echo •cbr />cbr />cbr />c°tsr />cbr />';
do_htlll_footerC);
```

Veamos los pasos de este listado. En primer lugar, iniciamos una sesión y comprobamos si el usuario se ha conectado como administrador, ya que no queremos que nadie m'5 pueda cargar archivo&- Estrictamente hablando, tambiin deberiamos comprobar si en las variables l i • t ymai lid hay caracteres no deseados, pero lo hemos omitido por motivos de brevedad. Seguidamente, definimos y enviamos

los encabezados de la página, y comprobamos que el formulario se haya completado correctamente. En este caso es muy importante, ya el formulario resulta dificil de completar para el usuario. Tras ello creamos una entrada para este correo en la base de datos y creamos un directorio en el archivo para almacenar el correo.

A continuación, la parte principal de la secuencia de comand0IJ, en la que se comprueba y se mueve cada uno de los archivos cargados. Esta parte es diferente si se cargan varios archivos. En este momento debemos procesar cuatro matrices, con losnombres$_PILES [•userfile •J [•name'),$_PILES [•uaerfile') [•tmp_name' J, $_PILES ['userfile •J ['size •] y $_PILES [•userfile •J (•type •]. Se corresponden a los equivalentes de mismo nombre de la carga de un solo archivo a excepción de que cada una es una matriz. El primer archivo del formulario se detallaen$_PILES ['userfile •J [•tmp_na- •J [Ol,$_PILES [•uaerfile •J [•name') [0),$_PILES ['userfile') ['aize') (0J y$_PILES['uaerfile'l ['type '1 [0J • Con estas tres matrices, realizamos las habituales comprobaciones de seguridad y añadimos los archivos al archivo principal.

Por último, le ofrecemos una serie de botones al administrador que puede utilizar para previsualizar el boletín informativo cargado antes de enviarlo, así como un botón para enviarlo. El resultado de upload. php se muestra en la figura 30.13.

Figura 30.13. La HCUencia de comandos de carp lnfonnll de los archivos cargllldos aal como de su tmmafto.

Vista previa del boletín informativo

El administrador ~pone de dos forrnu de acceder a una vista previa de un boletín informativo antes de enviarlo. Puede acceder a las funciones de vista previa

EL_HOLISTICO@HOTMAIL.COM - 2006

desde la pantalla de carga si quiere previsualizar el boletín inmediatamente. despufs de cargarlo. También puede pulsar el botón Vlew MaU. que le mostrará todos los boletines informativos sin enviar que hay en el sistema, para que los pueda previsualizary enviar más tarde. El botón VlSW Man activa la acción view-mail, que desencadena el siguiente código:

```
ca•• •vléw-mall' :
(
   dleplay_lt•••l'Uneent Mail', get_uneent_mail(get_email(II,
         'preview-bt•l', 'prevlew-text', •eencl•l,
   break,
```

Como puede comprobar, se utiliza de nuevo la función display _i tema () ~ botones para las acciones preview-html, preview~text y send

Un aspecto interesante que conviene destacar'es que los botones Preview no desencadenan realmente una acción, sino que realizan un enlace directo al boletín informativo en el archivo. Si se fija en los listados 30.7 y 30.8 apreciará que utilizamos la función display _preview_button () para crear estos botones, en lugar de la función display _button () habitual.

La función display _but ton () crea un enlace de imagen a una secuencia de comandos con parámetros GET cuando sea necesario; la función display _preview_ but ton () ofrece un simple enlace al.archivo. Este enlace aparecerá en una nueva ventana, que se consigue por medio del atributo target•new de la etiqueta de ancla HTML. El leSaltada de la vista pre,,ia de la venién HTML de un belelin informativo se representa en la figura 30.14.

Flgu,. 30.14. Una vista pr,"'8 de un boletln lnformetlvo HTML. con lm6genes •.

Enviar el mensaje

Al pulsar el botón Sencl de un boletín informativo se activa la acci6n send, que desencadena el siguiente código:

```
çaae 'aend' :
{
   ••ndl$_0ET('i~'l, $_SSSSIOIIC'.admin_uaer•]) !
   break;
```

De esta forma se invoca la función send {), incluida en la biblioteca de funciones mlm_fri•. php. Es una función bastante extensa y en la que utilizaremos la el~ Mail_mi-. El código de nuestra función se muestra en el listado 30.17.

LlstNo 30.17. Funcl6n sencl() de mlm_fns.php. Estll funcl6n envla un boletln informativo.

```
functlon ••ndl$..illd, $admin_u.er)
{
   if(!cbeck adllin ueer($admin u.erll
      retur_n-fal••;-
   lf(l($info • load_..11_info($..ilidlll
   (
      ecbo •cuu,.ot load U•t infor,ution for .....ge $..Uid"z
      i:iEui:iCfiiiei
   1
   S•ubject • $info[0];
   $liatid • $info(1J;
   S•tatu• • $info(2];
   S•ent • $infol3J;

   $from_na_ • 'Pyr .. id NLIC';

   $from addre•• • •returneaddr•••'/
   $query. ••elect email from IN!l_li•t• wbere lletid • $li•tid"1

   $conn • db connect(l1
   $re•ult • $conn->query($query1;
   if 11$rHultl
   (
      ecbo $query;
      return fal•el
   }
   el•• lf ($re•ult->n1111_row.• •O!
   {
      ecbo •Tber~ l• nobody •ub•cribed to li•t number $li•tid";
      return false;

      /I ineluya el•••• de eorreo de PBAR
      require('Nail.pbp');
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
rec¡uire('llla_il/1lille.pllp'l1

// iutam:ie una claH NIIIII: y ptMla  al carlcter da ratono  da carro/
// 11118Ya linee utiliaado ea eate aiat ...
.......e•  .... 11a11_a1..c•\r\a•1,

// lea la verai6D ea tato del boletla iafo__ti_
StatfUe-    • •arcbive/SUatid/S—Uid/tut. tzc• /
Stfp.  fopealStutfile ..... •r•1,
Stut • fread(Stfp, fileal..CSt..tfil.....111
fcloH (Stfpl,

// lea la veraim lffllL del boletla iaf—tiw
Sbtlll.file-    • •arcbive/Sliatid/Smailid/indez.bcmJ.•1
Sbfp • fa,ea1Sllt191fll.....,  •r•1l
Sbtal • Cread(Sbfp, Cileaiae(Sbtalflle11&8ell1
fcl08eCSbfpl ı

// dada irna. y tato  al cmjato al-u
s..aaage->aat'1'1T8ody(Statl1
S..aaagÍt->Htlfflll1OCly(Sbelll.) ı

// cmteaga la U:ata da illAgenaa nudonadaa coa Hta —-:1•
Sqwwry • •aalect patb, a1.. typa fr-  laagaa vbara •iUd • s-ilid";
Sranlt • Scoaa·>q11&ry(Sqwwryl,
UIISraault1
(
   ecbo •cp>UD&l:ab ee ,-C. imap IUt. fro11 dat.abatl·«/p>.
   ratura talH ı
}
••- • Sr•alllt->a-_rova,
CorCSi • 0, SicS.....,, Si++I___ ___ ___

   //cargue toda• laa lm&geaaa daada dlaco
   Srov • Sranlt·>Catcb_array(l1
   Slagfila.... • •arcbive/Sliatld/S-ilid/•.srov[OJ;
   Siagtypa • Srov[1l;
   // aliada toda• laa lúgaua al cmjeto
   .....ga->acldl1'1'IILI...-CSiagUl—. Siagcypa, Siagfll ——, trual,


// cr•••l  cuerpo del ..a••:I•
Sbody • s-••ave·>99t(l1

// cr.. loa eacabeaadoa del .....:I•
,i~•  ••'.get_raa1_ ....csaclalA_uaar1.''  c•.s.clmia_uaar.'>';
Slldrarray • array(
      •Proa'      •> Sfrae ..
      '8ubject'   •> Saubjactl;

Qdra • S9"Nga-»llaad8n(Sbdraa:z,ayl,

// cr.. al objeto qww aa ,,.. a •vlar
Saeadar •• 11au, ,Cactory0 '..U.'1,

if(Satatu• • • •s,,,a111•1
```

```
// aavla al mauaja irna. al adallliatrador
Snader·>Hllll(Sadaia_UHr, Slldra, Sbodyl ı

// aavla la veraim  ea tato  del mauaja al adaiaiatrador
aall<Saclaia_uaar, Saubjact, Stut,  .•~•  ''
      .9et_raal_....1Sadaia_uaarl. '' e' .Sadaia_uaar.">"l ı

ac:llo •11aü aaat to s.dllia_uar•,

// -rqww el boletla ia f—tlw e-  leido
Sqiwry • •upcllca aaU  aat atatua • 'TBITSD' •llera —iUd • ...ilid";
Sraault • Scoaa·>q1Wry(Sqwwryl1

echo •cp>Pren  Hllll ..-111 to NIIII .. U to vllole Uat.c/p>cceDter>'1
diaplay_1NttoaC'HIIII', °614- ... ilid"I ı
ac:llo 'c/caater>ıl
1
alH  if(Satatua • • 'TS8TSD'I
(
   //eavle toda la liata

   Sqiwry • •aelect aub9crlbera.raal....,  aub_Uau .... U,
                    aubacribera.ai..typa
      cr-  •ub liats,  aubacriHrir
      vbar• li•tid • Sliatid alld
            aub_liata ....U • aubacribera....11•1
```

```
Sraault • Scoaa·>qiwryCSqiwryl1
U IISranltl
   echo '<P>~ror  ge~!~~ aubac,:~~~~latc/p>' ı

SC0llllt • 01
// para cada aucriptor
vbila( Saubacrlber • Sraault·>Cetcb_rov(II
(
   ifCSaubecriber[2J• •'K'I
   (
      //aavla una verai6D H'nlL a loa qua la aolicitea
      Saaadar·>•alldlSaubacrlber(ll,  Sbdra, Sbodyl1
   }
   •]••
   (
      //aavia una verai6D ea tato a loa qiw ao qularaa correo
      // IffllL
      —UCSaubacriber(ll,  Saubjact, Stat,
      •rraa, ''.gat_r ..1_....(Sadmia_uer).'"c'.Sadllia_uHr.•••)1

Sqaery • "update au.l aet atatu • 'SIIIT',  —t • -U
         vbara ..ilid • Saailid";
Srenlt • ·scoaa->qiwryCsqwwryl,                    -1
echo •cp>A total of s —— t -...-•  -re aaat.c/p>"f
```

```
elee ifl$etatue • • 'IIIIIT')
(
   ecbo 'cp>Thie ..11 hae already been eent.c/p>',
```

Esta fundón reáiiza diferentes operaciones. Remite pruebas del boletín informativo al administrador antes de enviarlo. Realiza el seguimiento por medio del estado de un correo en la base de datos. Cuando la secuencia de comandos de CUP. de archivos carga un correo, establece el estado inicial del mismo como •STORBD• (almacenado).

Si la función send (l encuentra un correo con el estado •STORID•, lo actualiza a •TBSTBD• (probado) y lo envfa al administrador. El estado •TBSTBD• significa que se ha mandado una prueba del correo al administrador. Si el estado es •nsTBD•, se cambia por •CHANGBD• (modificado) y se envfa a la lista.

Esto significa que todos los correos se envfan dos veces: una en modo de prueba y otra en modo real.

La fundón tambi4n envfa dos tipos de correo electrónico diferentes: la versión en textQ,que se envfa por medio de la fundón mai 1 ( ) de PHP, y la venión HTML, que se envfa por medio de la clase Mail_lllillle. En este libro hemos utilizado mail (I muchas veces, por lo que nos centraremos en el uso de la clase Mail_mime. Jl lo analizaremos la clase exhaustivamente sino que explicamnos c:ómo la hemos utilizado en esta aplicación.

En primer lugar incluimos los archivos de clase y creamos una instancia de la clase Mai1_mime:

```
// incluya claeaa da correo da PBAR
includal Nail.php'l1
includal'Nail/ai ... php'I,

// inetancia una el••• NINB y p6eela al car&cter da retozno da carro/nueva
// llnaa utilisado en eete eietaaa
S.. aaaga • na• Nail_at.. (•\r\n•),
```

Apreciará que hemos incluido dos archivos de clase. Pmteriormente en esta secuencia de comandos utilizaremos la clase Mail genérica de PEAR para enviar el correo. Esta clase se incluye en su instalación de PEAR. La clase Mai1_mime se utiliza para crear el mensaje en formato MIME que se enviará.

Tras ello leemos las versiones en texto y en HTML del correo y las aftadim~ a la clase Mail_mime:

```
// lea la varei6n en texto del bolatln informativo
$taxtfilenaae.   •archive/$11etid/$aailid/text.txt•l
$tfp • fopan($taxtfila11&88,   •r•);
Staxt • fraad1$tfp,  fileeise(Stextfilanaael),
fcloea($tfp);

// lea la varei6n BTNL del bolatln informativo
```



```
flttalfU-     • •archiva/Slietid/Saailid/illdax.htal.-,
tltfp • fopan(flttalH1-,      "r"l,
tllc.l  • freadl$hfp,  filaeisal$htalfilenaae))1
fcloaa(tltfpl,

// dada  ll'IIL y texto al objeto aiaeaail
...aeaga•>eatTllTllody($text)1
...••aga>eatlfflllLllodyl$htall1
```

Seguidamente cargamos los detalles de las imágenes desde la base de datos y realizamos un bucle por las mismas, aAadiendo cada una de ellas al correo que queremos enviar:

```
$111111 • $reeult->nua_rowe,
for(Si • D; $ic$nua, Si++)
(
   // cargue toda• lae ill&g-•  deada al dieco
   Srow • Sraeult->fatch_arrayl);
   Siagfilanaae  • •archlve/$lietid/Saailid/•.$row[DI,
   $iagtype • $row(lll
      // dada  todu  lu  111igenee al objeto
   $mHaga->addlmU.Iaagal$iagfil-.     $lagtypa,  $iagUlana•, true);
```

Los parámetros que pasamos a addBTMLimage (> son el nombre del archivo de imagen (tambi4n podríamos pasar los datos de la imagen), el tipo MIME de la imagen, el nombre del archivo de nuevo y true ruau indicar que el primer padmetro es un nombre de archivo en lugar de datos de vo (sl qWS1erimos paiit sólo datos de imagen, pasaríamos los datos, el tipo MIME, un panmetro vado y falH). Estos parámetros son ligeramente complicados.

A continuación será necesario aear el cuerpo del mensaje, antes de poder definir los encabezados del mensaje. El cuerpo se crea de esta fonna:

```
// cree el cuerpo del rmneaje
$body • S-eeage,.  >get <) ,
```

Tras ello podemos crear los encabezados del mensaje con una llamada a la función headera (I de Mail_mime:

```
// crea loe encabesadoa del —Aje
$froa • '''.get   real naae($adllin uear).••   c'.$aclain_uear.'>'1
$hdrarray • array(  -              -
             •rro11,      •> $froa,
             'Subjact•   •> Seub:lactl;
Shdra • Saaaeag&•>haadere!$hdrarray!1
```

Por último, ~pu& de configurar el mensaje, podemos enviarlo. Para llevarlo a cabo, será necesario instanciar la clase Mail de PEAR y pasarla al mensaje que hemos creado. En primer lugar, instancia~ la clase:

```
// crea el objeto qua••  va a enviar
$eendar •• Nail::factory(•aail'l1
```

EL_HOLISTICO@HOTMAIL.COM - 2006

(En este caso, el pariimetro ‘mai 1‘ indica a la clase Mai l que utilice la función mail () de PHP para enviar mensajes. También podríamos utilizar ' sendmail ' o ‘smtp‘ como valor de este parámetro con los resultados obvios.)

Seguidamente, enviamos el correo a todos nuestros suscriptores. Para ello, recuperamos y realizamos un bucle por todos los usuarios suscritos a esta lista, utilizando Mail send (l o la convencional mai 1 (l, en función de la preferencia de tipo MIME del usuario:

```
ifl$•ul:l•criber[ZI• •'H'I
|
   //uvi•  la ver•iOD HTML a lo• qua la hayan •olicitado ,
   $•andar ·>•aadl$•ul:lecribar(ll,  Shdr•, $bodyl,
}
•|••
|
   //uvia  la varaiOD en ~•xto a lo• qua AO quieran correo H'I'NL
   maill$•ul:lecriber[ll, $•ul:ljact,  Staxt,
         'FrOII:  ••.ga~_raal_na.. l$adllin_uaar1.•• c•.$adain_uaar.">"l1
```

~ primer parúnetro de $sender->Hnd () debería ser la dirección de correo electrónico del usuario; el segundo, los encabezados y el tercero, el cuerpo del mensaje. Eso es todo. Hemos terminado la creación de la aplicación de listas de correo.

An.,·pliar el proyecto

---- Coino süele ser habitual en estos proyectos, su hincionalidad se puecie ampliar de distintas formas, como indicamos a continuación:

- Confirmar la pertenencia con los suscriptores, para que nadie se pueda suscribir sin su permiso. Para ello, se suele enviar un correo electrónico a sus cuentas y se elimina a los que no respondan. Este enfoque tambim nos permite borrar de la base de datos todu las direcciones de correo incorrectas.
- Se puede permitir al administrador que admita o rechace a los usuarios que se quieran suscribir a sus listu.
- Se puede añadir funcionalidad de listas abiertas que permita a cualquier miembro enviar correo electrónico a la lista.
- Permitir que sólo los miembros registrados puedan ver el archivo de una determinada lista de correo.
- Permitir a los usuarios buscar listas que coincidan con determinados criterios. Por e;emplo, los usuarios pueden estar interesados en boletines informativos sobre golf. Si se llega a un número concreto de boletines informativos, la búsqueda permitiría localizar e;emplares concretos.

- Si tiene pensado utilizar una lista de correo de gnn tmldo, una aplicación PHP no es la mejor forma de enviar los menst;es Un gestQr de listas de como espedfico como ezm1m resultada mú ef:icu que la invocación repetida de mail (l en PHP. Evidentemente, siempre puede generar la interfaz de usuario en PHP, pero hacer que ezmlm se encargue del tramjo duro.

A continuación

En el siguiente capftulo implementaremos una aplicaci6n de foros Web que permitid a los usuarioe realizar debates en línea esbucturados por tema y cadenas de conversación.

EL_HOLISTICO@HOTMAIL.COM - 2006

Un método muy indicado para conseguir que los usuarios vuelvan a visitar nuestro sitio consisfe en ofrecer foros Web. Se puÑeft utilizar para cosas.tan variadas como grupos de debate sobre filosoffao como asistencia tKllica de.:p!O(fuctos. En este capítulo implementaremos un foro Web en PHP. Una posible alternativa consiste en utilizar un paquete ya existente como Phon:un para crear nuestros foros. Los foros Web en ocasiones se denominan tablones o grupos de debate encadenados. La idea de los foros es que los usuarios puedan at\adir artículos o preguntas y que otros usuarios las lean y las respondan. Cada uno de los temas de debate de un foro se denomina *aulnr11.* Implementaremos un foro Web con el nombre blah-blah que tendrá la fund~idad que describimos a continuación. Los usuarios podrán:

- Inidar nuevas cadenas de debate y at\adir artículos
- Aftadir artículos en respuesta a artículos existentes
- Ver artículos que hayan enviado
- Ver las cadenas de conversaddn del foro
- Ver la relación entre artfculos, es dedr, ver qué artfculos responden a otros

El problema

La configuradón de un foro plantea de hecho un interesante problema. Necesitaremos alguna forma de almacenar los artículos en la base de datos, con el autor, el

EL_HOLISTICO@HOTMAIL.COM - 2006

titulo, la fecha e información sobre el contenido. A primera vista, se asemeja bastante a la base de datos Book-0-Rama.

Sin embargo, el funcionamiento de la mayoría del software de foros se basa, además de en mostrar los artículos disponibles, en mostrar las relaciones entre artfculos. Es decir, podemos ver qué artículos son respuestas a otros artfculos (y de qué articulo provienen) y qué artfculos son nuevos temas de debate.

Encontrará muchos ejemplos de foros de debate que implementan estas características, como por ejemplo Slashdot (ht tp: / / slaahdot. org).

Debe planificar cuidadosamente la forma de mostrar estas relaciones. En este sistema, los usuarios deben poder ver un mensaje concreto, una cadena de conversación en la que se incluyan las relaciones o todas las cadenas de un sistema.

También debemos permitir que los usuarios inicien nuevos temas o que respondan. Será la parte fácil.

Componentes de la solución

Como hemos mencionado anteriormente, el almacenamiento y recuperación del autor y del texto de un mensaje es sencillo. La parte más compleja de esta aplicación es la bdsqueda de una estructura de base de datos que almacene la información que queremos así como una forma de desplazarnos eficazmente por dicha estructura. ~ estructura de los artfculos de un debate puede ser similar a la que mostramos en lafigura 31.1.

Figura 31.1. Un articulo en un debate encadenado puede ser el primer articulo de un nuevotema, pero normalmente es una rnpuestll a otro articulo.

En este diagrama, apredari que hay un envio inicial que empieza un tema, con tres respuestas. Algunas de las respuestas tienen sus propias respuestas. Útas podrían tener sus propias respuestas y así sucesivamente.

Mediante el análisis del diagrama podemos determinar la forma de almacenar y recuperar los datos de los artfculos as(como los enlaces entre los mismos. Este diagrama muestra una estructura de árbol. Si ya tiene experiencia en el mundo de la programación, sabrá que es una de las principales estructuras de datos que se utilizan. En el diagrama hay nodos, o artfculos, y enlaces o relaciones entre artfc:u.. los, al igual que en cualquier estructura de árbol (si no está familiarizado con este tipo de estructura de datos, no se preocupe, ya que analizaremos sus fundamentos sobre la marcha). Los trucos para conseguir que funcione son los siguientes:

1. Determinar la forma de asignar esta estructura de árbol al almacenamiento, en nuestro caso, en una base de datos MySQL.
2. Determinar una forma de reconstruir los datos según sea necesario.

Comenzaremos con la implementación de una base de datos MySQL que nos permita almacenar artículos, para lo que crearemos sencillas interfaces.

Al cargar la lista de artículos que se pueden ver, también cargaremos los encabezados de cada artículo en una clase tree node de PHP. Cada una de estas clases contendrá los encabezados de un artículo asf como un conjunto de respuestas para el mismo.

Las respuestas se almacenan en una matriz. Cada respuesta será a su vez un tree_node, que puede contener una matriz de respuestas a dicho artículo que, a su vez, son tree_node y asf sucesivamente. El proceso continúa hasta llegar a lo que se deDorninan noclas de hoja.deLárboL.los nodos que no tienen respuestas. De esta forma obtendremos una estructura de árbol similar a la de la figura 31.l.

Veamos algunos términos. El mensaje al que contestamos se denomina nodo principal del nodo actual. Todas las respuestas al mensaje se denominan nodos secundarios del principal. Le resultará más sencillo recordarlo si se imagina esta estructura de árbol como si fuera un árbol genealógico .

El primer artículo de esta estructura de árbol, el que no tiene nodo principal, en ocasiones se denomina nodo raíz.

Nota

Puede que no sea demasiado intuitivo, ya que nonnalmente dibujamos el nodo raiz en la parte superior del diagrama, al contrario que las ralees de un érbol real.

Para generar y representar esta estructura de árbol, escribiremos funciones recursivas (que analizamos en un capítulo anterior).

Hemos optado por utilizar una clase para esta estructura ya que es~ forma más sencilla de aear una estructura de datos compleja y que se pueda ampliar dinúnicamente pa~ esta aplicación. Al mismo tiempo, conseguimos un código sencillo y elegante para realizar operaciones de cierta dificultad.

EL_HOLISTICO@HOTMAIL.COM - 2006

Presentar la solución

Para comprenderrealmente lo que hemos flecho con este proyecto, lo DIÚ aconaejable es analizar ponnenorizadamente el código, como haremos en breve. En esta .aplicación, el código es menos extenso que en las anteriores, pero su complejidad es mayor. La aplicación cuenta únicamente con tres páginas reales. Tendremos una página de índice principal que muestre todos los artículos de los foros en forma de enlaces a los artículos. Desde aquf, podremos aftadir un nuevo artfculo, ver cualquier artículo de la lista o cambiar la forma en que se ve el artículo desplegarido o replegando lu ramas del úbo.l (como veremos a continuación). Desde la vista del artículo, podrá enviar una respuesta para el mismo o ver las que ya existen. La página de nuevo artfculo nos permite introducir un nuevo mensaje, ya sea una respuesta a un mensaje existente o un nuevo mensaje que no tenga relación con los anteriores. m diagrama de flujo del sistema se representa en la figura 31.2.

Figura 31.2. El sistema de fol'OI blah-blah consta de tres parlH prlncl~.

En la tabla 31.1 se recoge un resumen de los archivos de esta aplicación.

Tabla 31.1. Archlvo9 de la apllcacl6n de fol'OI Web.

index.pbp	Apllcaci6n	1.8 péglna principal que ver6n loa ueuarloa cuando accedan al lillo. Contiene una Usta de todos loa artlculoa del llllo, llsta que se puede abrir o cerrar.
new_poat.pbp	Apllcacl6n	Formularlo utUlzado para enviar nuevos artlculoa.
atore_new_poat.pbp	Apllcacl6n	Almacena artlculoa lntroducldoa en el for. nuarlo new_poat.pbp.
view_poat.pbp	Apllcacl6n	Muestra un tema canc:relO y una lleta de respuestas a dicho tema.
treenode_claH.pbp	BlbHoteca	Contiene la c1... treenode, que utilnremos para moalrar la jerarqula de los ~ mas.

Nombro	Tipo	Descripción
[illegible]	Biblioteca	[illegible] funclon• de la aplícacl6n (el Nilo de archivos de tipo bllloteca que eiümeramos).
data_valid_fna.pbp	Bllloteca	Funciones de v8lldacl6n de dalos•.
db_fna.pbp	Bllolecll	FunclonN de conecllvldad de baN de d• toa•
diacuaaion_fna.pbp	Biblioteca	Funclon• para almacenar y recuperar mensaje•.
output_fna.pbp	Bllloteca	FunclOnN para raprea..... HTML.
create_databaae.aql	SQL	SQL para definir la bese de dalos que necesitamos en la apllcacl6n.

A continuación nos detendremos en la implementación.

Diseñar la base de datos

Senl necesario almacenar una serie de atn"buro n o'lffcamnmu de los utieules que se muestren en el foro: la persona que lo haya escrito, el título del artfculo, cuAndo se ha enviado y el cuerpo del articulo. Por esta razón, necesitaremos una tabla de artfculos. Ciearemoa un Id. exclusivo para cada uno de los articulos, denominadopost id.

En cada artfculo será necesario incluir cierta informaci6n con respecto a su posición dentro de la jerarqwa. Podrfamos almacenar en el artfculo información sobre los nodos secündarios del mismo. Sin embargo, cada artículo puede tener numerosas respuestas, lo que puede provocar problemas a la hora de construir la bue de datos. Como cada articulo sólo puede ser respuesta de otro artfculo, resulta DWI sencillo almacenar una referencia al articulo principal, es decir, al artfculo al que responde este artículo. Por lo tanlo, disponemos de los siguientes datos para almacenar en cada uno de los artfculos:

- poetid: Un Id. exclusivo para cada artículo
- parent: El poet id del artfculo principal
- poeter: El autor de este artfculo
- title: m tftulo de este artfculo
- poeted: Pecha y hora del artículo que hemos enviado
- meesage: El cuerpo del artfculo

EL_HOLISTICO@HOTMAIL.COM - 2006

Añadiremos algunas optimizadones a toda esta informadón. Al tratar de determinar si un artículo tiene o no respuestas, tendremos que ejecutar una consulta para ver si hay otros artículos que lo tengan como principal. Necesitaremos esta información en todos los mensajes que mostremos.

Cuantas menos consultas tengamos que realizar, con mayor rapidez se ejecutará el código. Podemos suprimir la necesidad de dichas consultas si añadimos un campo que muestre si hay o no alguna respuesta. Este campo será c:hildre?J y será Booleano, es decir, su valor será 1 si el nodo tiene nodos secundarios y 0 en caso contrario.

Siempre hay que pagar un precio por las optimizacion~. En este caso hemos optado por almacenar datos redundantes. Como vamos a almacenar datos de dos formas, tendremos que prestar especial atención y aseguramos de que las dos representaciones coinciden entre sí.

Al ai'\adir secundarios tendremos que actualizar el principal. Si permitimos que se borren secundarios. tendremos que actualizar el principal para que la base de datos sea coherente. En este proyecto no vamos a incluir una función para eliminar artículos, por lo que evitaremos parte del problema. Si tiene pensado ampliar este código, recuerde este aspecto.

Conviene mencionar que algunas bases de datos nos serán de más ayuda que otras para nuestros propósitos. Si utilizamos Oracle, mantendrá la integridad referencial por nosotros. Si utilizamos MySQL, que no es compatible con désencadenadores ni restricciones de clave secundaria, tendremos que escribir nues-h'a, propias comprobaeieftes para 119eguraffl Cl8 de que lc,s datos tienertsentidocada vez que ai'\adamos o eliminemos un registro.

Realizaremos otra optimización más. Separaremos los cuerpos del mensaje del resto de los datos y los almacenaremos en una tabla independiente, ya que este atributo tendrá el tipo text de MySQL. Al tener este tipo en una tabla, se reduce la velocidad de las consultas a la misma. Como realizaremos numerosas consultas para generar la estructura de árbol, el proceso se ralentizará considerablemente. Si almacenamos los cuerpos de los mensajes en una tabla independiente, podremos recuperarlos cuando un usuario quiera acceder a un mensaje concreto.

MySQL puede buscar con mayor rapidez registros con un tamai'\o fijo que registros de tamai'\o variable. Si es necesario utilizar datos de tamai'\o variable podemos crear índices en los campos que se utilicen para buscar en la base de datos. En algunos proyectos, nos bastaria con dejar el campo de texto en el mismo registro en el que se encuentre todo lo demás y especificar índices en todas las columnas en las que realizaremos la bdsqueda.

Sin embargo, los ~dices tardan un tiempo en generarse y es muy probable que los datos de nuestros foros cambien continuamente, por lo que tendremos que volver a generarlos a menudo.

También ai'\adiremos un atributo area en caso de que posteriormente decidamos implementar varias conversaciones con la aplicación. En este caso no lo implemenfaremt,s ya que esta opción la reservamos para utilizarla más adelante.

Una vez realizadas todas estas consideradones, en el listado 31.1 se muestra el código SQL necesario para crear la base de datos de foros.

u.tado 31.1. create_databaae.sql. Código SQL para crear la base de datos de debate.

```
create databa•• diecueeion;

uae diecueeion;

create table beader
(
   parent int not null,
   poater cbar(20) not null,
   title cbar(20) not null,
   cbildren int default 0 not null,
   area int default 1 not null,
   poeted dateti.. not null,
   ponid  int uneigned not null auto_incre..nt pri-ry  key
);

create table body
(
   poetid int uneigned not null pri..ry key,
   •••&CJ•  text
);


grant eelect, i119ert, update, delete
on diecueeion.•
to diacuaeionelocalboet identified by 'paeeword';
```

Puede crear esta estructura de base de datos si ejecuta esta secuencia de comandos a través de MySQL como se indica a continuación:

```
myeql -u root: -p-c- creat:e_dat:abaae.eql
```

Tendrá que introducir su contrasei'\a raíz. También debería cambiar la contrasei'\a que hemos asignado al usuario.

Para comprender cómo se incluyen los artículos y las relaciones entre ellos en esta estructura, puede recurrir a la figura 31.3.

Figura 31.3. La base de datos cuenta con una estructura da árbol de forma relacional plana.

EL_HOLISTICO@HOTMAIL.COM - 2006

Como puede comprobar, el campo prindpal de cada artículo de la base de datos contiene el post id del articulo situado por encima en el árbol. El artículo prindpal es el artículo al que se responde.

Tambim–veráque el nódora~; postic!t,no tiene prindpal. Todos los nuevos temas de debate estarán en esta posición. En artículos de este tipo, almacenamos su principal como 0 en la b~ de datos.

Ver el árbol de artículos

A continuación, necesitamos algún modo de obtener informadón de la base de datos y de representarla en la estructura de árbol. Lo haremos en la página principal, index . php. Para los objetivos de esta explicación, hemos introduddo algunos envíos de ejemplo con ayuda de las secuencias de comandos new_post.php y store_new_post. php, que analizaremos en el siguiente apartado.

En primer lugar describiremos la lista de artículos ya que es la parte central del sitio. Tras ello, todo lo demú resultará más sencillo. En la figura 31.4 se muestra la vista inicial de los artículos del sitio que verá el usuario.

Figura 31.4.. La vista Inicia! de la llstll de artlculoa muestra |01 artlculoa de fonna "replegada".

Aquí vemos todos los artículos iniciales. Ninguno tiene respuestas ya que todos son los primeros artículos de un determinado tema.

Comprobará que disponelpos de una serie de opciones. Hay una barra de mend que nos permite lñadir un nuevo envío así como desplegar o replegar la vista de los artfculos.

Para comprender lo que significa, nos detenclmnos en los envíos. Algunos de ellos tienen sfmbolos mú (+). Significa que estos artfculos cuentan con respuestas. Para ver las respuestas a un determinado artfculo, basta con pulsar sobre dicho sfmbolo. El resultado se muestra en la figura 31.S.

Figura 31.S. La c:adena di! debate ~ penlltencla 18 ha delplegado.

Como puede comprobar, al pulsar sobre el símbolo más(+) se despliegan las respuestas al primer artículo y el símbolo se convierte en un símbolo menos(-). Si lo pulsamos, se repliegan todos los artículos de esta cadena, lo que nos lleva de nuevo a la vista inicial.

También comprobará que una de las respuestas tiene un símbolo más, lo que significa que cuenta con sus propias respuestas. Esto puede continuar de forma arbitraria y puede ver cada conjunto de respuestas si pulsa sobre el correspondiente símbolo más.

Las dos opciones de la barra de menlls, Expand y Collapse, permiten desplegar y replegar todas las cadenas, respectivamente. En la figura 31.6 puede ver el resultado de pulsar el botón BxpUld.

Si se fija atentamente en las figuras 31.S y 31.6 verá que hemos pasado detenninados parámetros de index. php en la línea de comandos. En la figura 31.S el URL es http://localhoat/phpmyaql3e/chapter31/index.php?expand•10#10.

EL_HOLISTICO@HOTMAIL.COM - 2006

Figura 31.8. Todas las cadenas se han desplegado.

lii,ia seeueneia ée<c,manckl& lo lee COIRO "De&pleganl- elemento oon el poslid 10'". EL símbolo # es simplemente un ancla de HTML que desplaza la página hasta la parte que se acaba de desplegar. En la figura 31.6, el URL es ht tp: / / localhoa t / phpmysqlle/chapterll/index.php?expand•all.

Al pulsar el botón Bxpuui se ha pasado el parámetro expand con el valor al 1.

Desplegar y replegar

Si analizamos la secuencia de comandos index.php (véase el listado 31.2) sabremos cómo se lleva a cabo esta opera_c:ión.

Listado 31.2. lndex.pllp. Sec:uencia de comandos pera crear la vista del articulo en la péglna princ:ipal de la apllcación.

```
dphp
   incluit.    e• tac:lude_fa•. pbp • l ;
   ••••ioa_etartCl1

   // ca.pruebe •i ee ha creado la variable de •e•i6n
   if (l i8Nt (t _SU8ICIII['expuaded'J))
   {
      $_SUSIOIIl 'expaacled•J • array0 ;


   // campruebe 91 H ha pulHclo un bot6n de de9Pliegue
```



```
   // expand debe ••r igual a •all' o a un poetid o no eetar configurado
   ifliaaetl$_GET('expand'III
   {
      ifl$_GBT('expand•J • • 'all'I
         expand_alll$_SBSSIOJl('expanded'));
      el••
         $_SBSSI09('expanded') ($_GET('expand')) • true;


   // COlll)ruebe 81 H ha pulHdo un bot6n de repliegue
   // collapee puede eer igual a all o a un poatid o no eetar configurado
   ifliaaetl$_GBT('collap•e'lll
   {
      ifl$ GBT('collap•e'I••'all'l
         $:szssIOHC'expanded•J • array0 ;
      elee
      uneet($_SBSSIOR('expanded') ($_GET('collap•e•JJ);


   do_html_header('Diecuesion Poete');

   dieplay_index_toolbarll;

   // mueetre la vers16n en Arbol de lae convereacionee
   diaplay_treel$_SBSSIOll('expanded'));

   do_htlll_footerl);
?>
```

La secuencia de comandos utiliza tres variables para realizar su trabajo, como indicamos a continuación:

- La variabie de sesión expanded, que realiza el seguimiento de las cadenas desplegadas. Se puede conservar de una vista a otra, para poder disponer de múltiples cadenas desplegadas. Esta variable es una matriz asociativa que contiene el postid de los artículos cuyas respuestas se muestran desplegadas.
- El parámetro expand, que indica a la secuencia de comandos qué nuevas cadenas se deben desplegar.
- El parámetro collapse, que indica a la secuencia de comandos qué cadenas se deben replegar.

Al pulsar un símbolo más o menos, o los botones Bxp9nd o Collapae, se invoca de nuevo la secuencia index. php con nuevos parámetros paraexpand o col lapse. Utilizamos expanded de una página a otra para saber qué cadenas deben desplegarse en una vista concreta. Inicialmente, la secuencia de comandos inicia una sesión y añade la variable expanded como variable de sesión en caso de que- no se haya hecho previamente. Tras ello, la secuencia de comandos comprueba si se ha pasado un parámetro expando col lapse y modifica la matriz expanded en consecuencia. El código del parámetro expandes el siguiente:

EL_HOLISTICO@HOTMAIL.COM - 2006

```
ifli•••t!$_G&T[•expaad•J)l
(
   ifl$_G&T[•ezpaad•) • • •all')
   expand_all1$_SS18XOJl('expanded')l1
   $_8U8XOJI('expanded' 1 ($_G8T['expaad' 11 •·true,
```

Si hemos pulsado el bot6n 'Bxpud. ie invoca la funci6n expand_all (I para ailadir todas las cadenu que tengan respuestas a la matriz expanded. como veremos mú adelante. Si qui&Wramosdesplegar una cadena concreta, tendrfamos gue pasar un post id a travá de expand Para reflejarlo,ailadimos una nueva entrada a la matriz expanded La función expand_all (1 se muestra en el listado 31.3.

Ustado 31.3. Función upand_..l() de dlacu1alon_fn1.php. Procesa la malrlz$expended para d8Ipl9g1lr tod• I• cade1111 del foro.

```
   // marque todae lae cadena• con ..cunclarioe cuando •• •••tren
deeplegadae
   $conn • db_connectl)1
   $query • 'eelect poetid fr- beader wbere cbildren • 1'1
   $reeult • $cODD·>q11ery($query)1
   $nu• • $reeult>nua row•1
   for(Si ~-JII-I.i!.~7 ti++>
   (
      $tbie_row • $reeult·>fetcb_row1)1
      $expanded($tbie_row(O))•true;
```

Esta función ejecuta una consulta de base de datos para determinar qué cadenas del foro cuentan con respuestas, como se aprecia a continuad6n:

```
eelect poetid fr- beader wbere c:bildren • 1
```

Todos los artfculos devueltos se alladen a la matriz e•nded ~jecutamos esta consulta para abonar tiempo mú adelante. Bastarfa con ailadir todos loa artfculoa a la lista desplegada, pero sería inútil intentar procesar respuestas que no existen.

El repliegue de artfculos fundona de forma similar pero al contrario, como se indica a continuad6n:

```
iflieeet1$_G&T('c:ollapee')))
(
   ifl$_G&T(•coll11P••'1• •'all')
   S_BUSIOJI('upended '1 • array11 1
   el••
   uneet 1$_SUSIOJI('ezpaaded•) ($_G&T(•c:ollapee•J)) 1
```

Puede eliminar elementos de la matriz expanded si anula su configurad6n. Eliminamos la cadena que se va a replegar o anulamos toda la matriz si toda la

p6gina se va a replegar. Todas atal openiciones son de preprocesamienlo,por lo que sabemos qué artfculos 1e deben mostrar y c:úles no. La parte principal de esta secuencia de comandos es la lnvocad6n de display_tree($_SBSSIOR ℓ'expanded.)), que genera iealmente el úbol de artfculos mostrados.

Mostrar los artículos

A continuadón analizaremos la función dbplay _tree (), que mostramos en el liatado 31.4.

Listado 31.A. Func16n dlsplay_trN() de oulpul_fns.php. Cree el nodo ralz de la eslruc:lln de *bol.

```
func:tion dieplay_tree($ezpended, -$row • 0, $etart • 0)
(
   // -•tre la vieta en &rbol d8 lae -n•cionee

   global $table_widtb1
   echo •ctable width-\~$table_width\•>1

   // C011Pruelle ei ..... tra toda la lieta o una eublieta
   U($etart>Ol
   $IIU!1liet • truei
   •]••
   $eubliet ~ r-.l~•]
   // conetruya la eetructura de trbol para repr..ent~r el re•ulita de lae
   // convereacionee
   $tree • new treenode($etart, '', '', '', 1, true, ·1, $expandlld; $eüliat) 1

   // indique al trbol que•• 11Ueetre
   $tree·>dieplay($row, $•üliet)1

   ec:bo 'e/table>';
```

El papel prindpal de esta funci6n consiste en crear el nodo rafz de la estructura de úbol. La utilizaremos tanto para mostrar el fndice completo como para aear subúboles de respuestas en la p6gina view_poat .pbp. Como puede comprobar, adopta tres pañmetros. El primero, $expanded, es la lista de post id de artículos que se representan desplegados. El segundo, $row, es un indicador del número de fila que se utilizad para determinar loa colores alternos de las filas de la lista.

El tercer parúnetro, r.start, indica a la función dónde debe empezar a mostrar loa artfculoa. Se trata del postid del nodo rafz del úbol que se va a crear y a representar. Si lo mostrúamos todo, como hicimos en la p6gina principal, serfa 0 (cero), lo que significa que se IIIOStrarfan tocios loa artículos sin principal. Si este parúaelro es 0, establecemos $subli•t como falH y mostramos todo el árbol.

Si el panmetro es mayor que cero, lo utilizamos como nodo rafz del úbol que se va a mostrar, definimos $sublist como true y generamos *y* mostramos sólo

EL_HOLISTICO@HOTMAIL.COM - 2006

parte del itbol (loatilizaffllOS en la NClaenda de comandoavi-_po•t. pbp). Lo más importante que hace esta funci6n • inatanciar ana inatancia de la clue treenode qae aepaesenta)a rafz del úbel. Realmente no • an utfcalo pero acb1a como principal de todos los uticulas de pl'íiller nivel. que no tiemn principiL Una vez COMtraido el irbol, buta con invocar su fanci6n de 1ep1Wiltati,6n para mostrar la lista de utfcalos.

Utilizar la clase treenode

El c6digo de la clase treenode se ncop en el listado 31.S (en capítulos Ulteriores se describe el funcionamiento de las clases).

Llatado 31.S. ClaM treenade de treenode_lna.pllp. Parte prtncip9l de la -,lc8ci6n.

```
c?pllp
// la• funcioaea para carqar, coaatrv.ir y
// repreaentar el Arbol •• ·~~entran en ••t• archivo

cla• treenode
{
   /.I C:acla nodo del Arllol ti•=• variable• •iembro que conc:1-
   // todO• loa dato• de un ce=aa:I• a excepci6n del cuerpo
   yar Sa_poetid,
   ~ar S•_title1
   ur se ft08Cer,
   wus•:ioetedr
   ,,.r"$• cllUdren
   Yar ) •:cllildliac,
   Yar S•_deptll,

   pablic function ~conatrv.ctl$poatid, Stitle, Spoater, Spoeted, Schildren,
            Saxpand, Sdaptll, $mipalldad, Seubliet1

      // al coaatrv.ctor claHn• la• variable• miembro, pero
      // crea recn&raiv..-.ce !aa partee inferiorea del &rbol
      Stllia·>•_ll08tid • Spoet!~:
      Stlli••>m title • Stitle,
      Sthia·>•:i;o.c•r • Spa9ter:
      Stllia·>•J'OIICed • Spoated,
      Stllia·>m clü.ldren •klliildr-1
      Stllia·>m:cllildluc • acayU,
      Sthia·>m_deptll • Sdeptll,

      // a61o aoa i11POrta lo que aparece por debajo de •ate Dado ai
      // ti- ••Cllllldarioa y H ha urcado para Hr daaple,ado
      // lu Rblleta• aiempre H deapU.egan
      ifCCS9áUacflSupudl u ~cb:-l
      (
         Sc:oan • cll1_coaaect(l1

         S.-r,- • •Hlec:t • fra. .... r ldlen pannt • ·Spoaeid arder bi, po9ted•
         Srenlt • $a.m- >q119ryjSqa,ary),
```



```
for ($C01111t•01 Srow • e.yaql_faccll_array(Sreaultl, ,COUDt••I
{
   iflS9áli•tl ISupandedl sr-1•poetid'I J • • tEWII
      $expand • true,
   .1 ..
      $apand • falH
Stllia·>m cllildliat(ScountJ • - treeaode(Sraw(•poatid'I,Srow[•title• J,
      -                          Srow(•poacer•J,Sraw[•poeted•J,
                                 sr-l'children'I, Sexpand,
                                 Sdeptll+l, Snpen:kd, $9Ülietl,
```

```
function diaplay($row, Saubliat • fal••>
{
   // coao ••tratada IIII ob:l•to, •• encazp da —•erar• a el '1-

   // Sr- noa indica en qut fila noa encantr-•
   /I para eaber qu• color 1• correaponde

   /I Saubliat - indica ai Ht- en la pagina principal
   //o•• la da •nujea. Laa p&ginaa da -DN:l• deberlADincluir
   // Saubliat • true.
   // 111 una aabliata. todo• loa -ajee H daapliegu y no hay
   // alllboloa •+•o•-•.

   // 81 •• eraea dal nodo da raiz vaclo, 110 lo —•acre
   lHsttib·>a_dipth>·i:l
   {
      //fila• de coloree aleernoa
      ecbo •cer>ced bgcolor•"';
      1f C$raw12l
         echo ·lcccccc•>·;
      •]••

      // el aa119rado indica la profundidad da la Htruccura da anidaciOa
      for(Si • 0, $ic$thia·>•_deptbl Si++I
      (
         ecllo 'd .. acc.•i.agea/epacer.gif• ——·u•
               vidth•"22" alt••• valign•"llott—•…',


      /I —••ere • o ·, o un ••paciador
      1f l l$9áliat ..., Stlli• ·-_cllildren W. n-flSthi• ·>m_cllildí.i9Cll
      // eae- •n la pagina prīacipal, con algwl09 aecundarioa, tadaa
      // daeple,ado•
      {
         // ai -d dHplegaclo. ofrezca un botOa para replepr
         echo 'ca hnf•"icdllx.pllp?collapae•'·
                     Stllia·>m_poatid.'l'.Selli• ·>m_paetid.··
            >cimg arc•"il1&9ea/ai.Jlull.gif" valt..-."1Iottam•
            heigh~··22• vidth•"22" alt••COll.... 'ftlrmad• border-•D•></a>';
```

```
•l•• if(IS•'Ubli•t  •• $thi• ·>a_children)
(
   // •i ••t& replegado, ofresca un botOn para de•plegar
   echo 'ca href••index.php?expaad•'·
      $thia·>•.JIC111tid. 'I' ,Sthi•·> ~ atid. '°>ciag nc••i-.ea/plua.gif•
      beigbt••22• width•"22• alt•"kpand Tbreail•&or~r-•o•>c/a>,1
}
el••
(
   // no bay ••cuadarioa o •• trata de una a'Ubliata, no ofresca niagOa
   // botOn
   echo 'cimg •rc••imagea/apacer.gif•  haigbt••22• widtb••22•
            alt••• valign••bott011•>•;


ecbo • ca nama • Stbia->m_poatid >ca bref.
   •vtew_poat.php?poatid•Stbi•·>•.JIO•tid'>Stbia·>•   title
   Stbia->m_po•t•r ' '.reformat_date(Sthia->a_poetedl.'c/a>';
echo 'c/td>c7tr>';

// incre..nte el contador de fila• para alternar coloree
Srow++;
}
// invoque la repreaentaciOn de todo• lo• ••cundario• de eete nodo
// un nodo •Olo tiene aecundarioe en eu lieta" ai aeta deaplagado
$nua_children • aiseofl$thie->a_childliatl1
for(Si • 0; Sic$nua_children; Si++)

$row • Sthi•·>m_childliat[$il·>diaplay($row,    Saubliet);

retum  Srow,


};

?>
```

Esta clue contiene la funcionalidad que controla la atista de úbol en la aplicación. Una instancia de la clase treenode contiene detalles sobre un determinado envio y enlaces a todas las respuestas de dicha clase. Esto nos proporciona las siguientes variabl~ miembro:

```
p'Ublic Sa_poatidi
p'Ublic S• title;
p'Ublic sm:sioeter;
p'Ublic Sm_poeted;
p'Ublic S• children,
p'Ublic sa:childli•tl
p'Ublic Sa_deptb;
```

Comprobani que t reenode no contiene el cuerpo de un artfc:ulo. No es necesario cargarlo hasta que el usuario acceda a la secuencia de comandos view_post. pbp. Tendremosque hacerlo con ~rta rapidez ya que estamos realizando una extensiva

manipulación de datos pua IIIOllrar la liata de úbol y tendnmos que volver a calcular cuando se actualiza la p6gina o cuando se pulsa un bot6n.

El modelo de notaci61lde eslu variables utiliza un modelo muy empleado en aplicaciones 00: las variables empiezan por •- para recordamos que se trata de variables miembro de lac:lw. La mayoria de estas variables se corresponde directamente a filas de la tabla beader de nuestra base de datos. Las excepciones son $m_cbildÍist y $m_deptb. Utilizaremos la variable $m_cbildlist para almacenar las respuestas a este artfmlo. La variable $m_deptb almacenani el número de niveles de úbol en el que nos encontramos, que utilizaremos para aear la representación. La función de construcd6n determina los valores de todas las variables, como se indica a continuación:

```
p'Ublic functioa ~conatruct(Spoatid, $title, $poetar, Spoated, Scbildren,
            $expaad, Sdepth, $UpUlded, $e'Ubli8tl

   // el conatructor define laa variable• aieéro, pero
   // crea recuraivaaente laa parte• inferior•• del &rbol
   $thi•·>a_poatid  • Spoetidr
   $thia·>•  title  • Stitle;
   $thia·>a:sioater  • $poeter;
   Stbia·>a_poated • $poated;
   $tbi•·>• cbildren •$cbildrell;
   Sthia->a:childliat  • arrayll,
   $thi•·>m_depth • Sdeptbr
```

Al construir el treenode rafz a partir de display_tree () de la página principal, lo que estamos aeando en nfiltctjd es Un nodb ftcttl'iu sh, • tfculw aociadell. Pasamos algunos valores iniciales:

```
$trae•  new treeaode($etart,  ••,      ••,  1,  true,  ·1,  $1!XP&ftded,  $a'Ubli•tl;
```

De esta forma se aea un nodo rafz con un $poatid cero, que se puede utilizar para buscar todos los envfos de primer nivel que tengan un principal cero. Definimos la profundidad como -1 ya que realmente este nodo no forma parte de la representación. Todos los envfos de primer nivel tendr4n una profundidad cero y se situanin en la i7.quierda de la pantalla. Los niveles inferiores siguientes se desplazan hacia la den!Cha.En este constructor, lo mú importante es que se instancian los nodos secundarios de este nodo. El proceso se inicia comprobando si resu~ta necesario desplegar los nodos secundarios. Solamente se llevará a cabo este proceso si un nodo tiene nodos secundarios y hemos optado por mostrarlos:

```
if((Se'UbliatlJSexpaadl   •• Scbildren1
(
   $cona•  db_coanect(I;
```

Tras ello nos conectamos a la bue de datos y recuperamos todos los envíos secundarios, como se indica a mntinuación:

```
Squery • •aelect • fr- header wbere pareat • Spo•tid order by poated•1
$re•ult  • $coaa-,,query($query1;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Seguidamente completamos la mab'iz $• childlht con instandaa de la clase treenode, que contienen las respuestas a losenvíos almac:mados en estetreenode, como se indica a continuación:

```
for ($C01111ta01 Srow a 9ay9e¡l_fetch_array($reeultl1 $COUDt++I
(
   if!Slllblietllsexpanded( Srow[•poetid'I I a a t%Uel
      Sexpand a true,
   elH
      Sexpand • f•l••1
   Sthi•·;.._cbildli•tCScountJa new treenocle(Srow[•po•tid'),Srow['titla•J,
            Srow[•poet•r•J,Srow[•poeted•J,
            Srow[•cMldreD'I,   Saxpand,
            Sdeptb+l, Sexp&llded, S•ubli•tl1
```

Esta última linea crea el nuevo treenode *y* sigue exactamente el mismo proceso que acabamoe de dacribir pero en el siguiente nivel inlerior. Es laparte recursiva. Un nodo de ,rbol principal invoca el constructor treenode, le pasa su propio post id como principal *y* aftade uno a su propia profundidad antes de pasarlo.

Tru ello se crea cada uno de los treenode, u(como sus propios secundarios, hasta que ya no haya respuestas o niveles que desplegar.

finalmente, invocamos la fundón de representadón del treenode raíz (de nuev~ en display_trae (l), como se indica a continuación:

```
Str••·>di•play(Srow, S•ubli•tl;
```

La fundón display (l comprueba en un principio si se trata del nodo raíz ficticio:

```
if!StM• ···-depth>·ll
```

De esta forma, podemos eliminar el nodo ficticio de la pantalla. Sin embargo, no queremos ignorar por completo el nodo raíz. No queremos que aparezca pero tiene que notificar a sus nodos secundarios que se muestren. Tru ello, la función empieza a crear la tabla que contiene los artículos. Utiliza el operador de módulo (1) para detenninu el color de fondo que tendr6 esta fila (de forma que sean alternos):

```
//fil.. de colore• alterDOe
echo •ctr>ctd bgcolor • '1
U 1Srowl2l
   ecbo ••1cccccc•»•1
alN
   echo ''tffffff'>°1
```

Seguidamente utiliza la variable **miembro $a_depth** pan determinar el sangrado del elemento~ Si se fija en lu figuras anteriores, comprobad que cuanto mú profundo es el nivel de una respuesta, mayor es el sangrado. Para ello, utilizamos lo siguiente:

```
/l el •...,r•do iDdica la profUlldidad de la Ht%Uc:tara de aDidaci6D
for(Si a 01 $ic$thi•·>a_depth1 $1++1
```



```
   ecllo 'ciag •rca•i•v••l•pac•r.gif• lleighta•22•
        Width-o22" alta•• valigDa•bottoa°>'I
```

La siguiente parte de la función determina si se incluye un botón mú (+) o menos(·),o no se incluye nada:

```
// —•tre +o-,   o IIII eapaciador
U '( |Saublbt H $thb•.a_childreD H aiseof|Sthh · .a_childliet)1
// ••t-   aD la p&gina priDcipal, hay algUDOe HCWIClari-. todo• deaplegadoe
{
   // d Htl deeplnado, ofresca un bot&l para reple,ar
   acllo 'ca hrefa•iDdea.php?collapna•.
                $thi•·•_po•tid.'l'.$tbi•·.a_po•tid.••
      >ciag •rca•~·va•l•inu•.gif• vali911a•bottae•
      lleiglta•22• widtha•22• alta•Collmpae Tllread" bordera•0•>c/a>'1
)
el.. if|IS•ubliat •• Sthi•·>•_childran1
{
   // ei Htl replegado, ofreaca IIII bot6D para deaplagar
   echo 'ca href • •indu.php?expaDd•'.
        Sthi•-.a.JI0llcid.•l•.Sthi•·.a_poatid. '°>ciag •rc·•iuva•/plu•.gif•
        haighta•22• width••22· alta•lxpmld Thread" bordara•0•>c/a>' 1
)
•]••
{
   // no hay HCalldari- o .. traca de wa aubli•t•1 ao ofrec..,• Ding-GD bot6n
   ecba 'ci .. eee •i..l••!mp•••s gif• ~efght,•22•vtdtb:e•22•
        alta•• valigDa•bottoa">'1
```

A c:ontinuadón, mostramos los detalles de esee nodo:

```
echo• ca nam a Sthi•···.JI0•tid >ca llref •
        'view_po•t.php?po•tid•Sthi•·.a_poatid'>Stbi•·.a_title
        $thi•·>•_po•t•r - '.reforaet_detelSthi•·>•_po•ced).•c/a~·,
echo •c/td>c/tr>';
```

Caaibiamos el color de la siguiente fila:

```
// 1—nteal cantador da fil•• para alterDer colore•
Srow++;
```

Tiu ello, todoa laa trHDOde e;ecatan1111 código. mduymclo elnodo rúz:

```
// iDvaqae la repre••taci&l o todoe loe aecmMlari- da qce Dado
// UD DOdo e6lo ti- NCllldarioa - ea lleta d utt. deeplegedo
Smia childr9D ª •i••of!Sl;hi•·.a childlietl;
for($i • O; $ic$DUm_cbildro; Si++I
(
   Srow a Sthie-,.._childliet (H.J->diaplay(Sraw, SnbliatJ",
)
retllrl1 Srow1
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Se trata de nuevo de una función recursiva que invoca todos los secundarios de este nodo para representarlos. Les pasamos el color de fila actual y conseguimds . que lo devuelvan cuando terminen, para poder realizar el seguimiento del color alterno.

Eso es todo para la clase. El código es bastante complej~~ede experhMntar·y probar la aplicación, y después volver a estudiarlo si está conforme con los resultados que genera.

Ver artículos individuales

La invocación de display_tree () nos ofrece una serie de enlaces a un conjunto de artículos. Si pulsamos sobre uno de estos artículos, accederemos a la secuencia de comandos view_post .php, con un parámetro del postid del artículo que queremos ver. En la figura 31.7 se muestra un ejemplo de esta secuencia de comandos.

Figura 31.7. Ahora podemos ver el cuerpo del mensaje de este envio.

Esta secuencia de comandos muestra el cuerpo del mensaje así como todas las respuestas al mismo. Comprobará que, de nuevo, las respuestas se muestran en forma de árbol, pero totalmente desplegadas y sin botones más o menos. Es el efecto de $sublist.

El código de view _post:. php se recoge en el listado 31.6.

Uataclo 31.1. vlew_poat.php. Muestra el cuerpo de un mensaje.

```
c?pllp
   // incluya biblioteca• de funcione•
   include ('include_fna.php'l1
   $poatid • $_GBT['poatid');
   // obtenga Ioa detalle• del 8811aaje
   $poat • get_poat($poatidl;
   do_htlll_header($poat['title')l1

   // aueatre el menaaje
   diaplay_poat($poat);

   // ai el menaaje tiene reapueataa, 11a6atrelaa en viata de Arbol
   if($poat[•children•J)
   (
      echo 'cbr /><br />';
      diaplay_repliea_line();
      diaplay_tree($_SESSION['expanded'), 0, $poatid);


   do_html_footer();
```

Esta secuencia de comandos utiliza tres llamadas de función principales para realizar su trabajo: get_post (), display_post() ydisplay_tree (). La fun· d6n get_post O obtiene los detalles de la función de la base de datos. En el liltado-31.7 puede ver su eédige.

Listado 31.7. Función get.JK)it() de discusslon_fns.php. Recupera un mensaje de la base de datos.

```
function get_poat($poatidl
(
   // extraiga un mensaje de la base de datos y devu6lvalo como matriz

   if(1$poatidl   return false;

   $conn • db_connect();

   //obtenga de 'header• toda la información de loa encabezados
   $query.  •select • from header vhere postid • $poatid";
   $result • $conn->query($queryl;
   if($reeult->num_rovs!•ll
      return false;
   $poat • $reault->fetch_assoc(I;

   // obtenga el mensaje del cuerpo y aft&dalo al resultado anterior
   $query • '.-lect • froia body vhere posti~ • $poetid";
   $result2 • $conn->queryl$queryl;
   if($result2->nwa_rovs>0I
   (
      $body • $result2->fetch_assocll;
      if($body)
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
        Spo•t(•me•••ge•) • $body('me•••ge'I;
    1
    return Spo•t,
```

Esta función, al asignarle un post id. ejecuta las dos consultas necesarias para recuperar el encabezado y el cuerpo del mensaje, y los combina en una sola matriz asociativa que devuelve como resultado. Tras ello, se pasan los resultados de esta función a la función display_post() de ouput_fns. pbp. Simplemente imprime la matriz con formato HTML, por lo que no la hemos incluido. Por último, la secuencia de comandos view_post . pbp comprueba si hay alguna respuesta para 1!5te articulo e invoca display_tree (1 para mostrarlas en formato de sublista, es decir, completamente desplegadas sin símbolos más o menos.

Añadir nuevos artículos

Después de todo esto, pasaremos a ver cómo se atlade un tema al foro. Un usuario puede hacerlo de dos formas: en primer lugar, si pulsa el botón New Poetde la página d~ fndke y, por otra parte, si pulsa el botón Reply en la página view_post •pbp. Ambu acciones activan la misma secuencia de comandos, new_post •pbp, pero con distintos.parámetros. En la figura 31.8 puede ver el resultado de new_post . pbp cuando se accede a esta secuencia de comandos tras pulsar el botón .Reply.

Figura 31.8. En las respuestasse allade y se marcaautomücamente el texto del origine!:

En primer lugar, ffieseenel URL:bttp: / /localboat/phpnyaqlle/cbapterll /new_post.pbp?parent•l8.

El parámetro pasado como parent será el pos tid principal del nuevo envío. Si pulsa New POllt en lugar de Reply, obtendrá parent•0 en el URL.

Seguidamente, comprobará que en una respuesta, el texto del mensaje original se ailade y se marca con un carácter >, al igual que en la mayoría de programas de lectura de correo o de noticias. Por último, de forma predeterminada el título de este mensaje adopta el título del mensaje original con el prefijo Re: . El código que genera este resultado se reproduce en el listado 31.8.

Ustado 31.8. new_posLphp. Permitea un usuario introducir un nuevo tema o responder a uno ya existente.

```
c?php
   inclwle l'include_f118.php'I;

   $title • S_POST('title•);
   Spo•t•r • s_POST('po•ter•J,
   Su•••~•• $_POST('•snge•);

   if(i..etlS_GBT('parent')II
      $¡18rent • $_GBT('parent'I;
   el••
      Sparent • $_POST('parent•];


   iflt$area>
      SUw• • I;

   if 11 Serrorl
   (
      if 1! SparentI
      (
         Sparent • o;
         if 11$t1tle!
            Stitle • 'New Po•t';

      ei.e
      (
         // obt•ng• el nombre del men•aje enviado
         Stitle • get_poet_title(SparentI;

         // adjunte Re:
         iflstrstr1$title, 'Re, '! • • fal•• 1
            Stitle • 'Re: '.Stitle;

         //compruebe que .i titulo encaj"a en la baH de dato•
         Stitle • sub8cr1Stitle, o, 201;

         //adjunte ün ••tilo de c:omill.. al mensaje al que ••t• r•spondiendo
         S.. •••9• • add_quoting(get_po•t_me•••g•ISparentlll
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
do_htal~b9ader1Stitle1,

di•pley_a.ew_po•t_foralSparent, Sarea, Stitle, S.. •••ge, Spo•t•rl:

if IserrorI
{
   echo •Your ....age wa• not •tored.
      Nalce •11re YoU bave filled in all fielde and try •9ain.•1


do_htal_footerl11
?>
```

Despu& de la configuración inicial, la secuencia de comandos comprueba si el principal es cero o no. Si es cero, se trata de un nuevo tema y requiere acciones adicionales.

Si se trata de una respuesta ($parent es el po•tid de un artfculo existente), la secuencia de comandos prosigue y define el título y el texto del mensaje original:

```
// obtenga el D01lbre del ..n••1• enviado
Stitle • get_po•t_titlelSparentl1

// adj11Dte Re,
ifletr•trlStitle,   'Re:  ') • • falee
   Stitle • •Re,  ' .Stitle;

//compruebe que el titulo encaja en la ba•e de dato•
$Eltl• • iUbatr($tltle,    o.  J0),

//adjunte un ••tilo de comilla• al ..n•aje al que ••ti re9poDdiendo
S.. •••98 • add_quoti119(get_po•t_..••age(Sparentl);
```

Lu funciones que utiliza son get_poat_title ‹›, get_post_meaaage(› Y add_ quot ing ( ) . Todas ellas se encuentran en la biblioteca de funciones discusaion_fna. php. A continuación las mostramos en los listados 31.9, 31.10 y 31.11, respectivamente.

Uat.do 31.1. Función get_post_tltle()de dlscusslon_fns.php. Recuperade la base de datos el titulo de un mensaje.

```
function get_po•t_title(Spo•tidl
{
   // extraiga de la baH de data. el noabre de un • n•aje

   ifllSpo•tidl  retllZ'D ''1

   ScoDD • db_connectl11

   //obtenga de 'header' toda la iDforaaci6a eobre la. ucabeaado•
   Squery • ••elect tttle froa b9ader where po•tid • Spo8tid";
   Sreeult • SCODD->query(Squeryl I
   if(Sre•ult->nua_row•1•11
      return ''1
```



```
tthi•_zaw • $re811lt->fetch_array(11
retlllrlI tthi•_ivw[OJ;
```

U... 31.11. Funcl6n get_post~mmmge() de cllsolmlon_fns.php. Recupere de la base de dalloa el cuerpo de un mensaje.

```
functiOD get_po8t_M88age($p08tid)
{
   // utrai9A de la baH de data el coli.taido cllllID -Aje

   if 1!Spa.tidl return '';
```

•–·•--to,

```
   Squ8ry. "Hlect ....mge froa body wbere poetid. Spo•tid•,
   $re811lt • $CODD·>query(Sc¡ueryl I
   iflSre.ult·>1111a_row•>0)
   {
      Sthi•_zaw • $re8Ult·>fetch_array(I;
      retum Sthi•_rowC0J,
```

Lu dos primeras funciones recuperan el enobezado y el cuerpo de ujJnensaje de la bue dedatos.

Lla1ÑO 31.11. Función edd_qüoiírig() de dlicil1ilcln_lnt.php. Sanu,a el 19ld! de IIII mensaje con slmbolos ">".

```
function 8dd_quotingIhtring,  Spattem • '> 'I
{
   // &a.da IID eetilo de c—Ulu para urcar el texto que aparecerA
   // utreco.illado en la re9Pµe•ta
   retum tpattem.•tr_repla~el"\n•, •\ntpett ~ ·· S•tri119J1
```

La fund6n add_quoting ( ) vuelve a aplicar formato a la cadena para que cada linea del texto original empiece con un símbolo, que de forma predeterminada es >.

Despué de que el usuario introduzca su respuesta y pulse el botónPolt, accede a la secuencia de comandos atore_new_poat. php. En la figura 31.9 puede ver un ejemplo del resultado de esta secuencia.

El nuevo tema aparece en la imagen. debajo de Re: ua ing gd? - Laura - 01: 09 07/07 /2004. Aparte de esto, la ¡,'gina es icMntica a la ¡,'gina index. php convencional. El código de atore_new_poat .php se muestra en el listado 31.12.

Uatado 31.1:Z. store_new_post.php. Allade el nuevo tema a la base de datos.

```
c?pllp
   iDclude 1•include_fn•.pbp'1 I
   iflSid • •tore_new_po•tlS_POST11
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
   include l'index.pbp•):
1
el••
(
$error•  true,
   include ('nev_poat.pbp'I:
```

Figura 31.9. Ahora se puede ver el nuevo tema en el árbol.

Como puede comprobar, es una secuencia de comandos breve. Su tarea principal consiste en invocar la función store _new_post ( ). Esta página carece de contenido visual propio. Si el almacenamiento es satisfactorio, veremos la página de índice. En caso contrario, volveremos a la página new_post . php para que el usuario lo intente de nuevo.

La función store_new_post () se muestra en el listado 31.13.

LlatNo 31.13. Función store_n-_post() de dlscussion_fns.php. Valida y almacena el nuevo tema en la base de datos.

```
fUDCtiOD atore_nev_po•tl$po•tl
(
   // velide y elmacene UD nuevo ..n•eje enviado

   $CODD • d!J_connectll;
```



```
// compruebe que no bey callpOS vacioa
ifltfilled_out1$poatll
(
   return fal•e;

$poat • clean_alll$po•tl:

//aaeg,lreae de que exiate el principal
if($poat(•pareat')!•OI
(
   $query • 'select poatid fr– beader wbere poatid • ''.$po•t['parent'J'''';
   $renlt • $c:o1ln->query!$queryJ;
   ifl$renlt->num _rov•l•ll
   (
      return fal•e;


11 compruebe que no••  un duplicado
$query • 'aelect  beader.poatid  from beader,  body vhere
         beader.po•tid • body.po•tid  and
         beader.pareat  • '.$post(•parent•l.'  aad
         beader.po•ter • ''.$poat['poater•J.''      and
         beeder.title  • ''.$po•t['titl••J.''     and
         beader.area  • '.$poat[•area•J.'    and
         body.me••age  • ''.$poat[•meaaa9eJ.'':
$re•ult • $conn->qu•ryl$queryl;
U  (l$re•ultl
(
   return fal•e,
1
if($reault->nua _row•>0I
(
   $tbis_rov • $renlt->fetcb_arrayll:
   return $tbia_rov(OI,

$query • 'inaert  into  beader value•
          ('' .$poat[ •parent•l. '',
           ••.$po•t['po•t•r•].••,
           '-. $poat('title•  1.•'.
           O,
           ••.$po•t.[•area•J      .'',
           nowll,
           IRILL
           )•;
$renlt • $cODD->query($queryl:
if  (!$rHultl
(
   return falae;


// fijeae en que ahora el principal tiene UD •eeundario
Squery • 'update beader ••t cbUdren  • 1 vhere poaeid • ''$poat['parent'l;
Srenlt  •  $conn->query($queryl;
if  !!$re•ultl
(
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
   retun  falaa,

// bueque el Id, del — j ⊖. Puede beNr  ver:I.0II encüeAdoa
// idtnticoa e uc:epci6n del Id. y J!.J:Pble•nte de la fecha de envio
$query • "Hlect  beader.poatid  fr:om beadar left jOID body.
         on beadar.poatid • body.poatid
         wbere parent • ''$poat ('pareat'J'''
         and poater • ••.spoat(•poater•J ....
         and title.  ••.spoat('title•J.••
         and body.poatid ia IRILL"1
$reault • ScODD·>queryl$queryl1
if  l1$rHultl
(
   return falae,
}
ifl$reault·>mai_rowa>0I
(
   $thia_rov • $reault·>fetch_arrayll1
   $id•  $tbia_row(Oll
}
if lSidl
(
   $query • 'inaert  into body veluea  ISid,  ••.spoatl'•••ege•J.·1•,
  $reault  • $CODD·>queryl$queryl1
   if  II $reaultl
   (
      return falaa,
   >

   return $idl
```

Es una fundón extensa, pero no excesivamente compleja. La longitud se debe a que para allaclir un tema es necesario ai\aclir entradas a lu tablas de encabezado *y* de cuerpo, uf como actualizar la fila del articulo principal en la tabla de encabezados para indicar que ahora tiene secundarios.

Con esto llegamos al final del c6cligo para la aplicadón de foros Web.

Añadir ampliaciones

Existen distintas ampliaciones que se pueden realizar en este proyecto:

- Se puede allaclir navegad6n a las opdones de vista, para que desde un tema se pueda acceder al siguiente mensaje, al mensaje anterior, al mensaje de la siguientecadena o al de la cadena anterior.
- Se puede allaclir una interfaz de administración para crear nuevos foros *y* aftadir temas antiguos.

- Se puede allaclir autenticación de uauarim para que solamente los usuarios registrados puedan enviar temu.
- Se puede allaclir alglin tipo de moderador o de mecanismo censor.

Puede recurrir a sistemas existentes si necesita nuevu ideas.

Utilizar un sistema existente

Existen algunos sistemas que conviene destacar. Phorum es un proyecto de foros Web de código abierto. La navepci6n *y* la aemúltica ION diferentes de la de nuestro sistema pero su estructura resulta butante sencillo personalizar para que se ajuste a la de nuestro sitio. Una de las can,cleristbs mú notables de Phorum es que el usuario lo puede configurar bien en vista con cadenas o en vista plana, Encontrarimúinformaciónalrespectoenladirecciónbttp://www.pborum.org.

A continuación

En el siguiente capitulo utilizaremos el formato de documento 'Pf>p para crear doaunentos atractivos, que 1e imprimande forma coherente *y* que, de alglin modo, os aplicar a una amplia gama ele aplicaciones buadas en servidos, como por ejemplo para génétll'c:vntralUS1, ...

EL_HOLISTICO@HOTMAIL.COM - 2006

Generar documentos personalizados en formato PDF

En sitios dirigidos por servicios, en ocasiones resulta necesario entregar documentos personalizados genera.dos en respuesta a entradas de nuestros visitantes. Se pueden utilizar para ofrecer un formulario completado de forma automática o para generar documentos personalizados, como por ejemplo documentos legales, cartas o certificados.

En el ejemplo de este capítulo presentaremos al usuario una página en línea de evaluación de conocimientos *y* generaremos un certificado. Nos centraremos en los siguientes aspectos:

- Utilizar procesamiento de cadenas PHP para integrar una plantilla con los datos de un usuario para crear documentos RTF (Fonnato de texto enriquecido).
- Utilizar un enfoque similar pan genenr un documento PDF (Formato de documento portable).
- Utilizar las funciones PDFlib de PHP para generar un documento PDF similar.

El problema

Nuestn intención es proporcionar a los usuarios un examen con una serie de preguntas. Si responden correctamente a la mayoría de las preguntas, genenremos un certificado que muestie que han aprobado el examen.

EL_HOLISTICO@HOTMAIL.COM - 2006

Para que un ordenador pueda calificarlos, las preguntas c:onstarát de. varias opciones, es dec:ir,estarát formadas por una pregunta y una serie de posibles res-. puestas. Sólo una de las posibles respuestasserli correcta.

Si un usuario obtiene una nota de apro~do,se le~arli un certificado.

El formato de archivo ideal de nuestro certificado deberla ser el sigujeJite:

1. Debería ser flic:il de diseftar
2. Debería incluir varios elementos diferentes, c:omoimlige~ ~ mapa de bits o vectoriales
3. La impresión del mismo tendrli que ser de gran calidad
4. Debería descargarse en un pequeftoarchivo
5. Debería generarse c:asi al instante
6. El c:ostede producciónde debería ser bajo
7. Debería funcionar en la mayoría de los sistemas
8. Debería resultar difícil de duplicar o modificar fraudulentamente
9. No debería necesitarse ningún software espec:ialpara verlo o imprimirlo
10. Debería mostrarse e imprimirse coherentemente para todos los receptores

Al igllal~ mucbasotraa .decisionn que debell!OSoma_r~ in~en~mos c:onseguir un formato que cumpla el mayor número posible de estos diez atributos.

Evaluar formatos de documento

La decisión mis ÍIÍlportante que debemos tomar es el formato que tendrli el c:ertific:ado.Entre las opciones posibles tenemos el papel, texto ASCil, HTML, Microsoft Word o el formato de cualquier otro procesador de texto, formato RU, PostSc:ripty PDF.Con los diez atributosenumerados !ftteriormente, podemos eemparar algunas de estas opciones.

Papel

La entrega del certiffc:adoen papel tiene algunas ventajas evidentes. Tenemos control total sobre el proceso. Podemos saber el aspecto exacto de c:ada certificado antes de enviarlo al receptor. No tenemos que preocupamos del software ni del anc:ho de banda y el certificado se puede imprimir con medidas antifalsificación.

Cumpliría todos nuestros requisitos a excepción del 5 y el 6. El c:ertiffc:adono se podría c:reary enviar de de forma rlipida. El envio postal podría demorarse durante días o semanas,en función de nuestra ubic:ac:ióny la del receptor.

Cada c:ertific:adotendrli unos costes de impresión, de franqueo y tambiál de entrega. La entrega electrónica automlitica resultaría mú ec:onómic:a.

ASCII

La entrega de doc:umenlaa en texto ASCU e texto sencillo tiene sus propias ventajas. La compatibilidad no ser6 un problema. El ancho de banda necesario no ser6 muy elevado, por lo que los costes serát bajos. La senc:illez del resultado final fac:ilitarli el disel\o y acelerad la c:reaciónde una secuencia de c:omandaa.

No obstante, si ofrecemos a nuestros usuarios un archivo ASCD, el control sobre el aspecto del certific:adoser6 mfnimo. No podemos controlar las fuentes ni los saltos de pligina. Solamente podremos incluir texto y no controlaremos el formato. No podremos evitar que el receptor duplique o modifique el documento. Es el método con el que resulta m6s f6c:il alterar fraudulentamente el c:ertific:ado.

HTML

Una opc:iónobvia para entregarun documento en la Web es HTML. El Lenguaje de marcado de hipertexto se ha diseftado eepedfic:amente para esta función. Como sin duda sabrli, cuenta c:on control del formato, sintaxis para aftadir objetos c:omo imligenes y es compatible (c:on algunos cambios) con distintos tipos de sistemas operativos y software. Es relativamente senc:illo,por lo que resulta f6cil de diseftar y las secuencias de comandos lo pueden generary entregarsin dific:ultad.

Entre los inconvenientes del uso de Hl'ML en esta aplicación podemos destacar la limitada compatibilidad c:on información ~e impresión c:omo los saltos de pligina, incoherencias a la hora de representarlo en distintas plataformas y programas, y una calidad de inq,lesi611nrieltle. Memú, aunque Hl'.ML puede incluir c:ualquier tipo de elemento externo, la capacidad de un navegador para representar o utilizai dichos elementos no se puede garantizar si se trata de tipos poco habituales.

Formatos de procesadores de texto

La entrega de documentos c:omo documentos de procesador de texto, en espedal en proyectos de redes internas, tiene bastante sentido. Sin embargo, en proyectos de Internet, el uso de un formato de procesador de texto propietario puede excluir a algunos usuarios, pero si consideramos su importancia en el mercado, Microsoft Word es una buena opción. La mayoría de los usuarios tiene acceso a Word o a procesadores de texto que puedan leer archivos de Word.

Los usuarios de Windows que no dispongan de Word pueden descargar el programa gratuito WordViewer de la dirección http://www.microaoft.com/office/000/viewera.aap.

La generación de un documento como documento de Microsoft Word tiene sua ventajas. El dlsefto es muy senc:illo, siempre que disponga de una c:opia de Word. El control sobre el aspecto impreso del documento es considerable y tambi4n disponemos de gran flexibilidad en lo que respecta a los c:on~dos. Tambim podemos impedir que el usuario modifique el documento si indic:amos que Word solicite una contrasella.

Desafortunadamente,los archivos de Word pueden tener un gran tamafto, sobre todo si contienen imligenes u otros elementos complejos. Tampoco resulta f6c:il

EL_HOLISTICO@HOTMAIL.COM - 2006

generarlos dinAmicamente en PHP. El formato est.i documentado, pero es un formato binario y la documentación de formato se incluye con las condiciones del contrato. Se pueden generar documentos de Word con un objeto COM, pero no es sencillo.

Otra posibilidad que puede tener en cuenta es OpenOfficeWriter, que tiene dos ventajas: no es un programa propietario y utiliza un formato de archivos basado en XML

Word 2003 también admite el formato de archivo XML La Definición de tipo de documento (DTD) para Word y para otros productos de Office se puede descargar de Microsoft.com. Es una opción vilida pero no resulta sencilla.

Formato de texto enriquecido

El Formato de texto enriquecido o RTF nos ofrece la mayoría de las ventajas de Word pero los archivos son ms fáciles de generar. Disponemos de gran flexibilidad sobre el diseño y el formato de la pilgina impresa. Podemos incluir elementos como imgenes vectoriales o en mapa de bits. Y tenemos la seguridad de que el usuario obtew.ldr6 un resultado similar cuando vea o imprima el documento.

RTF es urflonnato de texto de Microsoft Word. Se ideó como formato de intercambio para itransferir documentos de unos programas a otros. Tiene algunas semejanzas ~ Hn,tL. Utiliza sintaxis y palabras clave en lugar de datos binarios pera transmilir la información de formato. Por lo tanto, resulta legible. El formato estil bien doéumentado. La especificación estil disponible gratuitamente y la puede encontraren .. direcd6nhttp://msdn.microsoft.com/library/default.asp?url•/library/en-us/dnrtfspec/html/rtfspec.asp.

La forma tllú sencilla ele generar un d~umento RTF consiste en seleccionar la opción Guardar como RTF en un procesador de texto. Como los archivos RTF sólo contienen texto, se pueden generar directamente y los ya existentes resultan filciles de modificar.

Como el formato estil documentado y disponible gratuitamente, son mú los programas que pueden leer RTF que el formato binario de Word. No obstante, debe saber que los usuarios que abran un archivo RTF complejo en versiones antiguas de Word o en diferentes procesadores verm resultados muy distintos. En cada nueva versión de Word se aftaden nuevas palabras clave para RTF, por lo que las implementacione mú antiguas ignorarm los controles que no entiendan o que ,-.yan optado por no implementar.

En función de nuestra lista, resultaril muy sendllo diseftar un certificado RTF con ayuda de Word o de cualquier otro procesador de texto; puede contener distintos elementos como imgenes vectoriales o en mapa de bits; tiene una gran calidad de impresión; se puede generar sencilla y rilpidamente, y se puede enviar electrónicamente con un coste bajo. Funciona con diferentes tipos de aplicaciones y sistemas operativos, aunque con resultados variables. Entre los inconvenientes, diremos que ~ documento RTF lo puede modificar cualquiera, lo que resulta un problema para nuestro certificado y para otros tipos de documentos. En documentos complejos, el tamafto del archivo puede ser demasiado elevado.

RTF es una opción muy aconsejable para muchas aplicadones de envio de documentos, por lo que lo utilizaremos en este ejemplo.

PostScript

PostScript, de Adobe, es un lenguaje de descripción de pilginas. Se trata de un potente y complejo lenguaje de comunicación ideado para representar documentos independientemente del dispositivo, es dedr, una descripción que genera resultados coherentes entre diferentes dispositivos, como por ejemplo impresoras y pantallas. Está muy bien documentado. Existen varios manuales al respecto así como innumerables sitios Web sobre el tema.

Un documento PoslScript puede incluir formato de gran precisión, texto, iNgenes, fuentes incrustadas y otros elementos. Resulta muy sencillo generar un documento PostScript desde una aplicación si se imprime a través de un controlador de impresora PoslScript. Si le interesa, podría incluso aprender a programarlo directamente. Los documentos PostScript son bastante portables. Ofrecen impresiones de gran calidad en distintos dispositivos y sistemas operativos.

Sin embargo, a la hora de utilizar PostScript para distribuir documentos en~-tramos algunos inconvenientes:

- El tamafto de los archivos puede ser excesivo
- Los usuarios tendrm que descargar software adicional para utilizarlos

La mayoria de los uauario8 de Unix podrán procesu uchi.vos PastScript, pero los de Wmdows tendrm que descargar un visor como GSview, que utiliza el intirprete Ghostscript PoslScript.

Este software ~tá disponible para una amplia variedad de plataformas. Aunque es gratuito, no es aconsejable obligar a los usuarios a que descarguen programas adidonales, Encontrar, mals informació sobre Ghostscript .. http: //-w. ghostscript . can y lo puede descargar de http://www. es . wisc. ed/ghost. Para nuestra aplicación, PoslScript ofrece un resultado de gran calidad pero no cumple el resto de los requisitos.

Formato de documento portable

Afortunadamente, existe un formato con la mayoría de las prestaciones de PostScript, pero con algunas ventajas significativu. El Formato de documen~ portable (también de Adobe) se diseftó para distribuir documentos que se comportaran de forma coherente en las distintas plataformas, asf como para generar resultadoe de gran calidad tanto en papel como en pantalla.

Adobe describe PDF como "el est.indar abierto para la distribución de documentos electrónicos a escala mundial. Adobe PDF es un formato de archivo universal que conserva todas las fuentes, fonnato, colores y gr'6cos de cualquier documento original, independientemente de la aplicación y la plataforma u~das para aearlo. Los archivos PDF ~ compactos y se pueden compartir, ver, examinar e imprimir como desee cualquier usuario por medio de Adobe Acrobat Reader".

EL_HOLISTICO@HOTMAIL.COM - 2006

PDF es un formato abierto y la documentación correspondiente la puede encontrar enht tp: //partner•. adobe. com/aan/tech/pdf /apecif icationa. j sp, asf como en numerosos sitios Web y en el libro oficial.

Si lo comparamos con nuestra lista de mibutos, PDF parece muy indicado. Los documentos PDF ofrecen un resultado coherente de gran calidad y son capaces de incluir elementos como imgenes vectoriales y de mapa de bits, pueden utilizar compresión para crear un archivo mú pequefto, se pueden enviar elec:tr6nicamente y a bajo coste, se pueden utilizar en la mayoría de los sistemas operativos y pueden incluir controles de seguridad.

Uno de los inconvenientes de trabajar con PDF es que la mayoría de los ºprogramas utilizados para crear este tipo de documentos son comerciales. Se necesita un lector para ver archivos PDF pero Acrobat Reader se ofrece gratuitamente para Windows, Unix y Macintosh. Muchos visitantes de nuestro sitio estadn familiarizados con la extensión .pdf y probablemente tenga instalado el lector.

Los archivos PDF resultan muy indicados para distribuir documentos atractivos y que se puedan imprimir, en especial los que no queremos que se puedan modificar por parte de los lectores. Veremos dos formas distintas de generar un certificado PDF.

Componentes de la solución

Para que el sistema funcione, tendremos que examinar el conocimiento de los usuarios y, suponiendo que pasen la prueba, generar un certificado que informe de su rendimiento. Experimentaremos con tres formas para generar este certificado: dos con PDF y una con RTF. A continuación veremos con ms detenimiento los requisitos de cada uno de estos componentes.

Sistema de preguntas y respuestas

La creación de un sistema flexible de evaluación en línea qúe admita distintos tipos de preguntas, distintos tipos de medios para admitir la información, datos útiles sobre preguntu erróneu así como un acertado informe estadístico es una tarea de gran complejidad. En este capitulo, nos centraremos b,sicamente en el reto que supone generar documentos personalizados para enviarlos por la Web, por lo que únicamente crearemos un sencillo sistema de preguntas. El examen no depende de ningún programa especial. Utiliza HTML para realizar lu preguntas y PHP para procesar lu respuestas, lo que hemos hecho desde el primer capitulo.

Software de generación de documentos

No se necesita software adicional en el servidor Web para generar documentos RTF o PDF a partir de plantillu, pero necesitaremos algún programa para crear la

plantilla. Para poder utilizar lu funciones de creación PDF de PHP, tendremos que haber compilado la compatibilidad con PDF en PHP (como veremos en breve).

Software para crear una plantilla RTF

Puede utilizar cualquier procesador de texto para generar archivos RTF. Para crear nuestra plantilla de certificado hemos utilizado Microsoft Word. La plantilla de certificado se incluye en el correspondiente directorio del CD-ROM.

Si prefiere utilizar otro procesador de texto, es aconsejable que pruebe el resultado en Word ya que es el programa que utilizan la mayoría de nuestros usuarios.

Software para crear una plantilla PDF

Los documentos PDF resultan mú diffciles de generar. La forma mú sencilla consiste es adquirir Adobe Acrobat, programa que le permite crear archivos PDF de gran calidad a partir de diferentes aplicaciones. En este proyecto hemos utilizado Acrobat para ~r el archivo de plantilla.

Para crear el archivo, hemos utilizado Microsoft Word para diseñar un documento. Una de las herramientas del paquete Aaobat es Adobe Distiller. Dentro de Distiller tendremos que seleccionar algunas opciones no predeterminálu. El archivo se debe guardar en formato ASCD y es necesario desactivar la comJSrelión. Una vez definidu estas opciones, basta con imprimir el archivo PDF.

Encontruá mú informadón sobre Acrobaten la direcdónhttp://www. adobe. cam/produeta/aorob&t: y lo puede comprar en linea o en su proveedor de software habitual.

Otra opción para crear PDF es el programa de conversión ps2pdr que, como indica su nombre, convierte archivos PostScript en archivos PDF. Su principal ventaja es que es gratuito pero no siempre genera buenos resultados en documentos con imgenes o fuentes no estmdar. Este conversor se incluye con el paquete Ghoslsoipt que mencionamos anteriormente.

Evidentemente, si vamos a crear un archivo PDF de esta forma, primero habd que crear un archivo PostScript. Para ello, los usuarios de Unix disponen de lu utilidades a2ps o dvipa.

Si trabaja en un entorno Windows, tambiffl puede crear archivos PostScript sin Adobe Distiller, aunque a travá de un proceso mú complicado. Tendn que instalar un controlador de impresora PostScript. Puede utilizar, por ejemplo, el controlador Apple LaserWriter DNT. Si no tiene instalado ningún controlador PostScript, puede descargar uno de Adobe enhttp: 11-. adobe. com/aupport/ downloads /product.jsp?product•44•platform•Nindowa.

Para crear su archivo PostScript, tendm que seleccionar esta impresora y ejecutar la opción Imprimira archivo, que se incluye en el cuadro de dWogo Imprimir.

La mayoría de lu aplicaciones de Windows generan un archivo con extensión .prn, que deberla ser un archivo PostScript. Debería cambiar el nomine del archivo a .ps. Lo podr' ver por medio de GSview u otro visor PostScript, o crear un archivo PDF con ayuda de la utilidad ps2pdf.

EL_HOLISTICO@HOTMAIL.COM - 2006

Debe saber que los distintos controladores de impresión generan resultados PostScript de calidad variable. Seguramente descubra que algunos de los archivos PostScript dan problemas cuando se ejecutan con la utilidad ps2pdf. Le sugerimos que utilice un controlador de impresora diferente.

Si solamente tiene pensado crear un número reducido de archivos PDF, puede que le interese el servicio en línea de Adobe. Por sólo 9,99 dólares al mes, puede cargar archivos en distintos formatos y descargar un archivo PDF. Este servicio fue suficiente para nuestro certificado, pero no le permite seleccionar opciones que son importantes para el proyecto. El PDF creado se almacenará como archivo binario y se comprimirá. Por esta razón resulta muy complicado mod!ficarlo. Puede encontrar este servicio en http://createpdf.adobe.com.

Existe una opción de prueba gratuita que puede utilizar.

También puede encontrar una interfaz gratuita para ps2pdf basada en ftp en Net Distillery, http://www.babinszki.com/distiller.

Una última opción consistiria en codificar el certificado en XMLy utilizar Transformaciones de hoja de estilo XML (XSLn para convertirlo a PDFo a cualquier otro formato. Este método requiere conocimientos de XSLT y no lo describiremos en este libro.

Software para crear PDF mediante programación

, PHP incorpora compatibilidad con la creación de documentos PDF. Existen diferentes bibliotecas de funciones, con objetivos similares. Las funciones PDFlib de PHP utilizan la biblioteca PDFlib, disponible en http: //www.pdflib.com. Las funciones ClibPDF utilizan la biblioteca ClibPDF, disponible en http://www.fastio. com.

Ambas bibliotecas son similares. Proporcionan un API de funciones para generar un documento POF. Hemos optado por utilizar PDFlib ya que parece que se actualiza con mayor regularidad.

Conviene mencionar que ninguna de estas bibliotecas es software gratuito. Ambas permiten cierto uso no comercial sin cargo, pero si tiene pensado o&ecer un servicio comercial que las utilice, tendrá que abonar una cuota.

Empiezan a aparecer algunas bibliotecas gratuitas, como FPDF, aunque no es tan completa como las bibliotecas comerciales. Además, como está escrita en PHP (y no en C como extensión de PHP), no resulta tan nipida como las demás. Puede descargarla en ht tp: / /www. fpdf . org.

Para ver si POFlib ya está instalada en su sistema, compruebe el resultado de la función phpinfo (). Por debajo del tftulopdf veni si se activado la compatibilidad con PDFlib, así como la versión que se utiliza.

Si tiene pensado utilizar imágenes TIFF o JPEG en sus documentos PDF, tendrá que instalar también la biblioteca PDF, disponible enht tp: / /www. lihtif f . org/ y la bibliotecaJPEG, que encontrará enftp://ftp.uu.net/graphics/jpeg.

En un sistema Unix, PDFLib cuenta con bibliotecas de objetos compartidas para PHP 4.3 y 5.0. Se pueden cargar desde php. ini o con dl (1, por lo que lo es necesario volver a compilarlo para añadir esta funcionalidad.

En un servidor Windows, la DLL PDFlib se incluye en el archivo PHP Zip, por lo que basta con anular el comentario de la extensión en su archivo php. ini.

Presentar la solución

Crearemos un sistema con tres posibles resultados. Como se indica en la figura 32.1, realizaremos una serie de preguntas, evaluaremos las respuestas y, tras ello, generaremos un certificado de una de estas formas:

- Generaremos un documento RTF a partir de una plantilla en blanco.
- Generaremos un documento POF a partir de una plantilla en blanco.
- Generaremos un documento PDF mediante programación por medio de PDFlib.

Flgu111 32.1. Nuestro sistema de certificación generará uno de estos tres certillcados.

En la tabla 32.1 se recoge un resumen de los archivos del proyecto de certificación.

Tabla 32.1. Archivos de la aplicación de certlllcaclón.

Tires	Oil	Spark Plugs
index.html	Pjgina HTML	Fannulario HTMl. que contiene las preguntas de la pn,etle
score.php	Apllcaci6n	Secuencia de comandos para ewluar las respuestas de los usuarloa
rtf.php	Apllcaci6n	Secuencia de comandos para genenir un certfflc:ado RTF a partir de una plantilla

EL_HOLISTICO@HOTMAIL.COM - 2006

pdf.pbp	Apllcaci6n	Secuancla de comancloa para generar un "'bertlllcado PDF a partir da una planllla
pdflib.pbp	Apllcacl6n	Secuencia de comandos para generar un cerUflcado PDF por medio de PDFllb
eignat11re.png	Imagen	Imagen de mapa de bits de la firma que se Incluiré en el certlftcado PDFllb
PHPCertifieation.rtf	RTF	PlanWla de certlflcado RTF
PHPCertifieation.pdf	PDF	Plantilla de certificado PDF

A continuación analizaremos la aplicación.

Responder a las preguntas

El archivo index. html es muy sencillo. Debe incluir un formulario HTMLque pregunte el nombre del usuario así como las respuestas a una serie de preguntas. En ~ aplicación de evaluación real, probablemenle tendríamos que recuperar estas preguntas de una base de datos. En nuestro caso nos centramos en la creación del certificado, por lo que codificaremos espedficamente algunas preguntas en el código H'fML.

El campo name es un campo de entrada de texto. Cada pregunta cuenta con tres botones de opción para que el usuario indique la pregunta que desee. El formulario tiene un botón de imagen y un botón de envío.

El código de esta página se recoge en el listado 32.1.

Uatado si.1. Index.html. P6glna HTML que ~llene las preguntas de la prueba.

```
chtal>
   cbody>
      chl>cp align••cantar°>
          ciag arc••roaatta.gif•   alt•"">
          Cartificati0II
          ciag arc••roaatta.gif•   alt•"°>c/p>c/hl>
      cp>You too can aarn your highly raapectad PBP cartification
      fr0II tha world faaoua Pictioaal lnatituta  of PRP cartificatioa.c/p>
      cp>Siaply anawar tha quaatioaa bal-•c/p>

      cfora action••acora.pbp•   aathod••poat">

         cp>llhat doaa tha PBP atat...nt echo do?c/p>
         col>
            cli>cinput typa••radio• naaa••q1• valua•"l">
               Outputa atringa.c/li>
```



```
            cli>cinput typa-"radio• naaa••q1• valua••Z•>
               Add8 two llllllllara togatbar.c/li>
            cli>cinput  typa-•radio• naaa••q1• valua••3•>
               Craataa a l199ical elf to finiab writing your coda.c/li>
         e/oh

         cp>llbat. doaa tba nP fuctioa coa()  do?c/p>
         col>
            cli,cinput  typa-•radio~ ......•q2•  value•"l">
               calculataa a coaina in radiane.c/li>
            cli>cinput typa••radio• ......•qz• valua•°Z°>
               Calculatae a tangent in radiana. c/li>
            cli>cinput type-•radio• naaa••qz• valu••"l">
               lt ie nota  PRP functioa it ie a lattuca. c/li>
         c/Ob

         cp>lfbat doae tbe IIIP functioa ..u()  do?c/p>
         <Ol>
            cli>cinput typa••radio• naaa••q3• value•"l">
               Senda•  uil aaeNga.
            cli>cinput typa•"radio• naaa••q)• valua•"Z">
               Cbacke for ••  uil.
            cli>cinput type-•radio• nama•"q3• valua•"3">
               Togglea PRP bet-  ula  and f—1• aoda.
         e/oh

   cp align••canter">cinpat typa••i..ge• arc••certify-m.gif•  borde~•o•>c/p>

   t/18.lH•
      c/body>
   c/ht11h
```

Alcargar index. html en un navegador Web,se produce el resultado que mostramos en la figura 32.2

Calificar las respuestas

Cuando el usuario envía sus respuestas a las preguntas de index. html, tendremos que calificarlas y calcular una puntuación. Para ello utilizamos la secuentja de comandos acore. php, cuyo código se muestra en el listado 32.2.

Uatado S2.2. acore.php. Secuencia para cal~r los exémenea.

```
c?pbp
   //crea n0llbrae da variable• cortoe
   $q1 • $_1'0STl'q1'¡
   $qZ • $_f0S"!-'l'qZ'¡
   $q3 • $_1'0STI'q3'J ı   •
   $naaa • $_POSTl'naaa•J ı

   // comprueba qua ee baD recibido todo• loa datoa
   UC$qlo  •"ll$qZ•  •"ll$q3•  •"ll$naaa•  •"I
   (
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
ec:bo •cb1>ep aligD • c-ter:.d.99 arc-•ro•ette.gif• alt•••,.
                                  &orry:
                                  eimg arc••ro..tta.gif• alt•••:.e/p>e/hl>';
ecbo 'cp:.You naed to Ull in your - and an-r all quaationac/p>';

•l••
(
 ||•_ l•• puatu.ac:ione•
S•cor• ' 0,
1f(Sq1 •• 1) // la raapue•t• con:ecta da ql aa 1
   S•cora++I
1f 1Sq2 • • 11 // la reapue•ta correcta da qZ aa 1
   S•cora++;
if(SqJ • • 1) // la reapueata correcta da qJ aa 1
   0 0 ••,

//c0111'lierta la P11Dtuaci6n an un porcentaje
S9COra • sacora / J • 100,

1f (S.-      e SO)
f
   /I, aata peraona ba auapendido
   aebo •ebl ali111••canter">ei119 arc••ro..tta.gif• alt••• />
                                  SOrry:
                                  ei119 arc••roaatta.gif• alt••• />e/bl>'1
   ac:llo 'ep>YOU naad to acore at laaat 501 to pea• tba - __/P>';
)
•l••
(
   // crea mia cadana que contenga la puntuaci6D con U11 decimal
   Sacora • nUllber_format(Sacore, 1);
   -...•..
         •eb1 align••canter•>ei119 arc••ro..tte.gif• alt••• />
            CODgratulat1011•
            eimg arc••roaatte.gif• alt••• />e/bl>'1
         •ep:.Nell done Sna.., witb a acore of Sacorel,
         you bawe paand tbe —— .e/p>°;

   /I illcluyaulacaa a •ecuenciu da ~ ·      que ganeru loa cartificadoa
   ec:llo •cp:.Plaa.. click bar• to downloed your certificata u
      a Ni~ft !lord (Rff) Ule.c/p>' 1
   ec:llo •cfora action••rtf.p!Jp• ..tbod••poat">'1
   ec:llo 'ecenter>
      eillput typa••image• arc••certificate.gif• bordar••O•>
      e/cantar>';
   eclIo •einput cype• •taiddea • ua1le••acore• value•• ' . S•core. '•>• ;
   ec:bo •ciapu.t type••llidden• na-··DaM• value••• .S..-. '•>
   eclIO 'e/fora> '1

   ec:llo •ep>Pl.._. click bere to dowllload your certificata aa
         a Portable ~t      Po:niat (1'DPI fila.e/p>';
   ec:llo 'efora action••pdf.p!Jp• ..tbod••poat">' 1
   ac!IID •ecaatar>
      cillPUt typaa•i.age• arc••certificate.91~· bordar••O•>
      e/cantar>'1
   ecllo 'e.input. cype.•b:Lclclen• aame•••core• ,ralu .. •• . Sacor•. '•,. •:
```



```
ecbo 'eillpu&typa-~!Jiddq•  ___- -___.  val~· · -_-, ••,.,.,
ecllo 'e/form>';

ecbo 'ep>Pl•••• click llera to download your certificate ••
   a Portable Doc:umallt Po:niat (PDP) file generated witll l'DPLib.e/
P>':
ecbo 'efora action•°pdflib.p!Jp• •tbod••poat°>';
ecbo 'ecantar>
   cinput typa•"imaga• arc••cartificata.gif• bordar•"O°>
   c/center>';
echo '<input type••biclclea• na.. ••core• valu••••.$8COre.••,.•,
echo 'cinpu.t type••bicldea•  11a1119 .....      valu• ···.snaae.••>•;
acbo 'e/forra>';

?>
```

Figura 32.2. La ptgi1111 lnclu.hlml pideel .....io.que .-pandll a• preguntas de la pruebe.

Esta secuencia de comandos II\OIIInri un mema;e si el usuario no ha respondido a todu-las preguntas o si su puntuación es inferior a la calificac:ión de aprobado que hayamos seleccionado.

Si el usuario respon~ correc:taall!ftte a las preguntas, podrá generar un certificado, como se muatra en la figua 32..3.

EL_HOLISTICO@HOTMAIL.COM - 2006

FIgura 32.3. La secuencia de comandos score.php ofrece a los visitantes que hayan realizado la prueba correctamente la opción de generar un certificado de tres formas.

A partir de aquí, el usuario dispone de tres opciones. Puede conseguir un certificado RTF o uno de los certificados PDF. Veremos cada una de las secuencias de comandos correspondientes a cada opción.

Generar un certificado RTF

No hay nada que nos impida generar un documento RTF si escribimos texto ASCII en un archivo o en una variable de cadena, pero para ello tendríamos que aprender otra sintaxis. Veamos un documento RTF muy sencillo:

```
{\rtU
{\fonttbl {\f0 Arial1}{\f1  Ti..a ''W Roaan,}}
\f0\fa2a  Heading\par
\f1\fa20  Tiiia ia an rtf docu..nt.\par
)
```

Este-dotumento establece una tabla de fuentes con dos fuentes: Arial, a la que se hará referencia como fo y Times New Roman, a la que se hará referencia como f 1. El encabezado utiliza fo (Arial) con un tamaño de 28 (14 puntos). El control \par

indica un salto de párrafo. Tras ello, escribimos Tbia is a rt f document (&te es un documento rft) en f 1 (Times New Roman) con un tamafto de 20 (10 puntos).

Podríamos generar este documento manualmente, pero PHP no cuenta con funci~_gue nos ~ tra~jo a la hora de crear las partes más complejas, como por ejemplo en la incorporación de gráficos. Afortunadamente, en muchos documentos, la estructura, el estilo y la mayor parte del texto son estáticos y solamente cambian algunas part~ de un !,lluario a otro. Una forma más eficaz de generar un documento consiste en utilizar una plantilla.

Con ayuda de un procesador de texto, podemos generar un documento complejo como el de la figura 32.4.

Figura 32.4. Por medio de un procesador de texto podemos crear una sencilla y atractiva plantllla.

Nuestra plantilla incluye marcadores de posidón como «HAMB» para indicar dónde se añadirán los datos dinámicos. El aspecto de estos marcadores no es relevante. Utilizaremos una descripdón con sentido dentro de las comillas angulosas. Es muy importante que seleccionemos marcadores de posición que no aparezcan de forma accidental en el resto del documento. También le ayudará a diseilar la plantilla si los marcadores de posición tienen aproximadamente la misma longitud que los datos por lo que los vamos a reemplazar.

EL_HOLISTICO@HOTMAIL.COM - 2006

Los marcadores de posición de este documento son cdfAMIb>, cdfame>>, «acore» y «—/dd/yyyy». Utilizaremos tanto NAME como Name, ya que emplearemos un método que discrimina entre mayúsculas y minúsculas para reemplazarlos.

Una vez disei\ada la plantilla, necesitamos una secuencia de comandos para personalizarla. Esta secuencia es rt f . php y su código se muestra en el listado 32.3.

LIs.. do 32.3. rtf.php. Secuencia de comandos para crur un certificado RTF personalizado.

```
c?php
   //cree nombres cortos de variables
   Sname • S_POST['name•J;
   Sscore • i POST[•score•J;
   // compruebe que di•pone de lo• parimetros necesarios
   ifl !sname || !Sscore I
   (
      echo '<hl>Error:c/hl>Thi•  page was called incorrectly';
   }
   el••
   (
      // genere lo• encabezados para que el navegador •eleccione
      // la aplicaci6n correcta
      header( 'content-type,  applicacion/msword' );
      headerl •content-Di•position,  inline, filena..•cert.rtf'I;

      $date • dateC 'P d, Y' 1;

      // abra el archivo de plantilla
      Sfil•—   • 'PRPCertification.r,:f•;
      Soutput • file_get_contents(Sfilsname);

      // sustituya lo• marcadores de posici6n en la plantilla por lo• dato•
      Soutput • •tr_replaceC '<<NANB>>',  strtoupper( Sname ), Soutpuc 1;
      Soutput • str_replacel 'ccName>>',  Sname, Soutput );
      Soutput • scr_replace( '<escore>>', Sscore, Soutput );
      Soutput • str_replace( 'cc..,./dd/yyyy>>',  $date, Soutput 1;

      // envle el doc:ulllento generado al navegador
      echo Soutput;
```

Esta secuencia de comandos realiza una comprobación de errores básica para garantizar que se han pasado todos los detalles del usuario y, tras ello, pasa a crear el certificado. El resultado de esta secuencia de comandos será un archivo RTF en lugar de un archivo HTML, por lo que tendremos que indicárselo al navegador del usuario. De esta forma, el navegador intentará abrir el archivo con la aplicación conecta o proporcionará un cuadro de diálogo Guardar como si no reconoce la extensión R~. Especificamos el tipo MIME del archivo por medio de la función header () de PHP para enviar el correspondiente encabezado HTIP como indicamos a continuadón:


```
headerc•content-type,  application/msword'I;
beader(•content-Disposition,  inline, filename•cert.rtf'I;
```

El primer encabezado indica al navegador que vamos a enviar un archivo de Microsoft Word (no es del todo cierto, pero es la aplicación que probablemente se utilice para abrir el archivo RTF). El segundo encabezado indica al navegador que muestre automáticamente los contenidos del archivo y que el nombre sugerido del mismo es cert. rtf. Es el nombre de archivo predeterminado que el usuario verá si trata de guardar el archivo desde el navegador.

Una vez enviados los encabezados, abrimos y leemos nuestro archivo de plantilla RTF en la variable $output y utilizamos la función str_replace () para reemplazar nuestros marcadores de posición con los datos que queremos que aparezcan en el archivo. La lfnea

```
Soutput • str_replaceC 'ccN—>>',  Sname, Soutput );
```

reemplazará cualquier presencia del marcador de posición «Name» por los contenidos de la variable $name. Después de realizar las sustituciones, basta con duplicar el resultado en el navegador. En la figura 32.5 puede v_er un ejemplo de esta secuencia de comandos.

Flguni 32.5. La secuenciade c:om8ndos rtf.php genera un cef1lflcado a partir de la plantilla RTF.

Este enfoque funciona a la perfección. Las invocaciones de str replade (I se ejecutan con rapidez, incluso a través de nuestra plantilla y, por tañto, los contenidos de $output son considerablemente extensos. El principal problema desde el punto de vista de esta aplicación es que @l usuario cargará el certificado en su procesador de texto para poder imprimirlo, lo que puede tentarle a que lo modifique. RTF no permite crear documentos de sólo lectura.

Generar un certificado PDF a partir de una plantilla

El proceso de generación de un certificado PDF a partir de una plantilla es muy similar. La principal diferenda es que al crear al archivo PDF, algunos de los man:adores de posidón se pueden mezclar con códigos de formato. Por ejemplo, si nos fijamos en el archivo de plantilla de certificado que hemos creado, veremos que el aspecto de los marcadores de posidón es el siguiente:

```
ccNl•ll(AMB)-10(>1·5(>
ccNal-,<•IOle)•lll>>
c)·lllcll(acl·l7(orl·&lel·&(>l·11(>
c)·ll(c)l(ml-12(ml0(/dl·&(dl·l,l/ll(yy)·ll(yy)·ll(>>
```

Comprobará que, a lo largo del archivo, al contrario de lo que sucede con RTF, no es un formato que nos resulte legible.

> El formato del archivo de plantilla PDF puede variar en función de la versión de Acrobat o de la herramienta de generación de PDF que utilice. Puede que el código incluido en este ejemplo no funcione cuando genere sus propias plantillas. Compruebe su plantilla y modifique el código para adecuarlo a la misma. Si sigue experimentando problemas, util1ce el ejemplo PDFUb que veremos més adelante. ...

Hay varias formas de solucionar este aspecto. Podríamos ir marcador por marcador y eliminar los códigos de formato. De hecho, apenas influye en el aspecto final del documento ya que los códigos incrustados en la plantilla anterior indican la cantidad de espacio que se debe dejar entre las letras de los marcadores que, de todas formas, vamos a sustituir. No obstante, si adoptamos este enfoque, tendremos que modificar manualmente el archivo PDF y repetirlo cada vez que cambiemos o actualicemos el archivo. No resulta complicado si sólo tenemos cuatro marcadores de posición, pero se puede convertir en una verdadera pesadilla si por ejemplo tenemos varios documentos con multitud de marcadores de posición y decidimos cambiar la letra en todos los documentos.

Podemos evitar este problema si utilizamos otra técnica. Podemos utilizar Adobe Acrobat para crear un formulario PDF, similar a un formulario HTML con campos en blanco. Tras ello, utilizamos una secuencia de comandos PHP para crear lo que se denomina un archivo FDF (Formato de ~tos de formularim) que büca· mente es un conjunto de dato& que se combinarán con una plantilla. Se pueden crear FDF por medio de la biblioteca de fundones FDF de PHP, en concreto la funciónfdf_create O para crear un archivo, la función fdf_set_value O para definir los valores de los campos y la función fdf_set_file (I para definir el archivo de formulario de planti]Ja asociado. Seguidamente se puede pasu este archivo al navegador con el torreSpORdiente tipo MlME, en este caso, vad. fdf y el complemento Acrobat Reader del navegador debería sustituir los datos del formu· lario.

Es la forma correcta de hacer las cosas, pero tiene dos limitaciones. Por un lado, asume que disponemos de una copia de Acrobat. Por otra parte, es compliado sustituir texto que está en linea en lugar de en campos de un formulario. Esto puede constituir o no un problema, en función de lo que queramos hacer. Hemos utilizado la aeación de PDF para aear cartas en las que era necesario cambiar numerosos elementos en linea. Los FDF no funcionan correctamente para este tipo de cosu. Si por ejemplo quiere completar el formulario automáticamente, como en el caso de un formulario de impuestos en línea, no senl problema alguno. Encontranl mú información sobre el formato FDF en el sitio de Adobe (http://partners. adobe.com/asn/developer/acrosdk/forms.html).

También debería consultar la documentación FDF del manual de PHP si tiene pensado utilizar esteenfoque(http://www.php.net/manual/eniref. fdf .php).

A continuación aplicaremos la solución PDF al problema anterior.

Podemos buscar y reemplazar los marcadores de posición de nuestro archivo PDF si reconocemos que los códigos de formato adicionales solamente están formados por guiones, dígitos y paréntesis y que, por lo tanto, pueden~ con una expresión regular. Hemos escrito una función, pdf_replace (), para generar automáticamente una expresión regular de coincidencias para un marcador de posición y reemplazar dicho marcador con el texto correcto.

Con algunas versiones de Acrobat, los marcadores de posición son texto sin procesar y podemos sustituirlos con str _replace (I, como hemos hecho~.

Aparte de esta novedad, el código para generar el certificado por medio de una plantilla PDF es muy similar al de la versión RTF. Esta secuencia de comandoa se muestra en el listado 32.4.

Uatado 32.4. pdf.php. Secuencia de comandos para crear un certificado PDF personalizado a trav61 de una plantilla.

```
c?php
   ••t_ti.._limitl 110 ); // eata aecuencia de COll&ndo• puede aer muy lenta

   //cree nombre• de variable• corto•
   $name • $_POST[•na..•J;
   $acore•  $_POST['•core');

   function pdf replacel $pattern, $replacement, $etring I
   |            -
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
   $len• •trlen( $pattern );
   $regex • ••;
   for ℓ $i • 0, $ic$len: $1++
   (
      $regex .• $pattern($il:
      if C$ic$len·l)
         $regex •• '(\1·?(0·91+\(l?';
   }
   return ereg_replace ℂ $regex, $replacement, $8tring l;


if(!$namell!$8corel
(
   echo 'chl>Brror:c/hl>Thi• paga va• callad incorrectly';

el•e
(
   // genere lo• encabezado• para que el navegador seleccione
   // la aplicaci6n correcta
   header( •content·Di•po•ition: filename•cert.pdf';
   header( 'content·type: application/pdf' );

   $date • date( 'F d, Y' l ı

   // abra el archivo de plantilla
   $filename • 'PHPCertification.pdf';
   $output • file_get_content•($filenamel;

   // •u•tituya lo• marcadora• de po•ici6n en la plantilla por lo• dato.
   $output • pdf_replace( '<<lfAME>>', •trtoupper( $na- l, $output l
   $output • pdf_replace( '<<Name>>', $name, $output );
   $output • pdf_replace( '<c•core>>', $•core, $output l;
   $output • pdf_replace( '<<mm/dd/yyyy>>', $date, $output· l;

   // envie el documento generado al navegador

   echo $output;
```

Esta secuencia de comandos genera una versión personalizada de nuestro documento PDF. El documento, que mostramos en la figura 32.6, se imprimirá correctamente en distintos sistemas y resultará muy complicado modificarlo o cambiarlo. Comprobará que el documento PDF de la figura 32.6 es prácticamente idéntico al documento RTF de la figura 32.5.

Un problema asociado a este enfoque es que el código se ejecuta lentamente debido a la expresión regular de coincidencias que necesitamos. La ejecución de las expresiones regulares es mucho más lenta que la de str_replace(), que utilizamos en la versión RTF. Si quiere que coincida un elevado número de marcadores de posición ~ quiere generar muchos de estos documentos en el mismo servidor, podría considerar enfoques adicionales. En el caso de una plantilla más sencilla, no supone ningún problema. El grueso de este archivo se corresponde a los datos que representan las imágenes.

Figura 32.11. La secuencia de comandos pdf.php genera un certificado a partir de una plantilla PDF.

Generar un documento PDF por medio de PDFlib

PDFlib se ha ideado para generar documentos PDF dinámicos a través de la Web. Estrictamente no forma parte de PHP sino que es una biblioteca independiente con un gran número de funciones que se pueden invocar desde una amplia variedad de lenguajes de programación. Existen vinculaciones de lenguaje disponibles para C, C++, Java, Peri, Python, Tcl y ActiveX/COM.

Desde PHP 4.0.S, oficialmente PDFlib GmbH es compatible con PDFLib, lo que significa que puede remitirse a la documentación PHP en http://www.php.net/manual/ref.pdf.php o descargar la documentación oficial de pdflib.com.

Una secuencia de comandos Hello World para PDFlib

Una vez instalado PHP con PDFlib activado, puede probarlo con un sencillo programa como el del ejemplo Helio World que mostramos en el listado 32.S.

LI•tado 32.5. testpdf.php. Ejemplo Helio World claiCO que utiliza PDFllb a través de PHP.

```
c?php
   //cree un documento pdf en memoria
   $pdf • pdf_new();
   pdf_open_file($pdf, '');
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
pdf_•et_iafo(tpdf, •creator•, •pdfte•t.pbp•)1
pdf_•et_info($pdf, "Author•, "Luke Welling and Laura Tboia8on•)1
pdf_•et_info($pdf, "Title•, •&ello World CPRP)")1

// la carta U8 e• 11" x 8.5• y hay apr-i•daMate    72 puato• por pulgada
pdf_bagin_page($pdf, 1,5*72, 11*721"i

// afiada UD • rcador
pdf_add_bookmark($pdf, 'Page 1', o, 0);

$font • pdf_findfont($pdf, 'Ti..• RomaD', 'bo•t•, 0);
pdf_•etfont($pdf, $font, 2,1,
pdf_•et_text_po•($pdf,50, 700);

// ••criba el texto
pdf_•bow($pdf,'Rello,worldl')1
pdf_continue_text(tpdf,•(..y. PRP)'11

// finalice el docu..nto
pdf_end_page($pdf);
pdf_c10.. ($pdfl1

$data•  pdf_get_buffer($pdf);

// genere lo• encabesado8 para que el navegador pueda •eleccionar
'// la aplicaci6n correcta
header('content-Type: application/pdf'I
beader(•content Di•po•ition, inline, filename•te•t.pdf'I;
baader(•content-Lengtb, '   •trlen($data));

// repre•ente el PDP
echo $data,
?>
```

Si esta secuencia de comandos falla, el error que probablemente vea es el eiguiente:

```
Patal error, Call to UDdefined fuaction pdf_new()
in c:\program file•\apacbe                    ...
group\Apacba\btdoc•\pbpmy•ql3e\cbapter32\te•tpdf.pbp
on line,
```

Significa que no ha compilado la exteneión PDFlib en PHP.

La inetalación es muy eencilla, pero algunos detalla cambian en función de las versionee de PHP y de PDFlib que utilice. Puede revisar lae sugerenciae incluidas en las notae de los usuarios de la página PDFlib, en el manual comentado de PHP.

Una vez ejecutada esta eecuencia de comandos en su sistema, es necesario analizar su funcionamiento.

Las líneas

```
$pdf • pdf_new();
pdf_open_file($pdf, ••)1
```

inicializan un documen~ PDF en memoria.

La función pdf_set_info () nos permite etiquetar el documento con un aeunto, un tftulo, un creador, un autor, una lista de palabra, clave y un campo personalizado definido por el usuario.

En nueetro caeo definimoe un creador, un autor y un tftulo. Los seie campos de información son opcionalee.

```
pdf_•et_info($pdf, •ereator•, •pdfte•t.pbp');
pdf_8et_info($pdf, 'Autbor', 'Luke Welling.&Dd- Laura Tboia8on•)1
pdf_•et_info($pdf, 'Title•, 'Bello World (PRP)');
```

Un documento PDF está formado por una eerie de página,. Para iniciar una nueva página, tendremos que invocar pdf_begin_page (). Al igual que el identificador devuelto porpdf_open(), pdf_begin_page () requiere lae dimeneiones de la página. Cada página de un documento puede tener un tamai\o diferente, pero a menos que tenga una buena razón para no hacerlo, conviene utilizar un tamafto de papel común.

PDFlib funciona en puntos, tanto para el tamai\o de lae página, como para localizar coordenadas en cada una de las páginae. Como referencia, diremos que A4 mide aproximadamente 595 por 842 puntos y que el papel de carta U.S. tiene 612 por 792 puntos. Esto significa que nuestra linea

```
pdf_begin_page($pdf, 1.5*72, 11*72);
```

crea una página en nueetro documento con el tamai\o del papel de carta U.S.

Noee nemsaáo que IHI documento PDF' sea silRplemeRteun documento que ae pueda imprimir. Se pueden aiiadir numerosas funciones PDF al documento, como por ejemplo hipervínculos o marcadores. La función pdf_add_outline (› aiiade un marcador al contorno del documento. Los marcadoree de un documento aparecenen un panel independiente de Acrobat Reader, lo que nos permite pasar de una sección a otra. Eeta línea

```
pdf_add_booaart($pdf, 'Page 1•. o, 01,
```

aiiade un marcador con la etiqueta Page 1, que hace referencia a la página actual.

Las fuentee dieponiblee en el eistema varían de un sistema operativo a otro e indueo de un equipo a otro. Para garantizar la coneistencia de los resultados, existe un conjunto de fuentee básicas que funcionan con todos los lectores PDF. Las 14 fuentee principalee eon lae siguientes:

- Courier
- Courier-Bold
- Courier-Oblique
- Courier-BoldOblique
- Helvetica
- Helvetica-Bold

EL_HOLISTICO@HOTMAIL.COM - 2006

- Helvetica-Oblique
- Helvetica-BoldOblique
- Times-Roman
- Times-Bold
- Times-Italic
- Times-Bolditalic
- Symbol
- ZapfDingbats

Se pueden atladir fuentes distintas a las de este conjunto, pero aumentar, el tamai\o del archivo *y* puede que no se acepten en función de la licencia que tenga de esa determinada fuente. Podemos selecdonar una fuente, su tamai\o *y* una codificación de caracteres, como indicamos a continuación:

```
$font • pdf_fiadfont($pdf, 'Tiae•-Raman', 'bo8t', OI;
pdf_setfont($pdf, $font, 2tl;
```

El tamafto de las fuentes se especifica en puntos. Como codificación de caracteres hemos seleccionado host. Los valores admitidos son winansi, builtin, 111acroman. ebcdic o hoet, cuyo significado se muestra a continuación:

- winanai: Utiliza IS08859-1 múcaracteresespecialesañadidosporMiaoaoft, como por ejemplo el símbolo del euro.
- builtin: Utiliza la codificación incorporada en la fuente; Se suele utilizar con fuentes *y* símbolos no Latinos.
- macroman: Utiliza codificación Mac Roman. Es el conjunto de caracteres predeterminado de Macintosh.
- ebcdic: Utiliza EBCDICcomo se hace en sistemas IBM AS/400.
- hoet: Selecciona autom6ticamente macroman en Mac, ebcdic en sistemas basados en EBCDIC*y* winand en los sistemas restantes.

S- no necesita incluir caracteres especiales, no es importante la codificación que esco,a.

Un documento PDF no es como un documento HTML o un documento de un plVCalldor de texto. De forma predetenninada, el texto no comienza en la parte supatar izquierda *y* 0uye hasta otras lineas segdn queramos. Tendremos que indicar ddnde se situarm todas las lineas de texto. Como ya hemos mencionado, PDF aliliu puntos para indicar ubicaciones. El origen (la coordenada x, *y* [0, 0)) se encumtra en la equina inferior izquierda de la p6gina.

~ nuestra página es de 612 por 792 puntos, el punto (50, 700) se encuentra a dos tercios de pulpda con respecto a la parte izquierda de la p6gina *y* a una pulgada *y*

un tercio con respecto a la parte superior. Para definir la posición del texto en este punto, utilizamos·

```
pdf_set_text_po•CSpdf, so, 700);
```

Por último, una vez configurada la p6gina, podemos escribir texto en la misma. Para aftadir un texto a la posición actual con la fuente actual, utilizamos pdf_ehow<>• La lfnea

```
pdf_•bawlSpdf,•Hello,world!'I;
```

ailadeel texto "Hello World!• a nuestro documento.

Para puar a la siguiente línea *y* escribir m6s texto, utilizamos pdf_continue_text (l. Para atladir la cadena •(saya PHPl•, utilizamos

```
pdf_cont1nue_text($pdf,' l••Y• PHP1'l;
```

La ubicación exacta en la que apareced dependm de la fuente *y* del tamafto que hayamos seleccionado.

Si en lugar de líneas o frases utiliza p6rrafos continuos, la función pdf_show_boxed (l le resultan de mayor utilidad. Le permite declarar un cuadro de texto *y* aftadir texto al mismo.

Despu~s de aii.adir los elementos a la p6gina, tendremos que invocar pdf_end_apge () de la siguiente forma:

```
pdf_end_page(Spdf1;
```

Una vez terminado el documento PDF, tendremos que cerrarlo por medio de pdf_close (). Al generar ~ ar~~i.vo tamlri.én~ necesario cerrarlo.

Lalfnea

```
pdf_clo•elSpdf1;
```

completa la generaáón de nuestro documento Helio World.

Ya podemos enviar el PDF terminado al navegador:

```
$data• pdf_get_buffer(Spdf1;

// genere lo• encabezado• para que el navegador puada •eleccionar la
apl1cac16D correcta
beader(•Content•Type: appl1cat1on/pdf'l;
beadar('Content•D1•po•1tion, inline; filenule•te•t.pdf'I;
headar(•Contant Length, ' . •trlan($datall;

// repre•ente el PDF
echo $data,
```

Tambim podríamos escribir estos datos en disco. Podemos hacerlo con PDFLib si puamos un nombre de an:hivo como segundo parúnetro depdf_open_file (>. Al cierre de esta edición, esta posibilidad daba problemas en Wind,ows (PHP 5.0.0). Si tiene que esaibir los datos en disco, puede hacerlo manualmente.

EL_HOLISTICO@HOTMAIL.COM - 2006

Este ejemplo se deriva del ejemplo de C de la documentadón PDFlib Y. debería servirle como punto de partida.

Sepa que en el manual de PHP, algunos de los parámetros de las funciones PDFLib que aparecen como opcionales sog obligatorios en determinadas versiones de PDFLib. El documento que queremos crear para el certificado es m4s complicado, ya que incluye un borde, una imagen vectorial y una imagen de mapa de bits. Con las otras dos lKnicas, alladimos estos elementos con nuestro procesador de texto. Con PDFlib, tendremos que alladirlos manualmente.

Generar un certificado con PDFlib

Para poder utilizar PDFlib, hemos tomado algunas decisiones. Aunque se pueda duplicar exactamente el certificado que utilizamos anteriormente, resultarfa mú costoso generar y ubicar manualmente todos los elementos que utilizar una herramienta como Microsoft Word para maquetar el documento.

Utilizaremos el mismo texto que antes, incluyendo el sello rojo y la firma en mapa de bits, pero no vamos a duplicar el borde tan complejo. El c6digo completo de esta secuencia de comandos se incluye en el listado 32.6.

Listado 32.8. pdftlb.php. Generación de un certificado por medio de PDFHb.

```
c?pbp
   // czee nombre da veriablee corto•
   $aame • $ POST('aa1le• JI
   $ecore • i_POSTl'•core•J1

   if 1l $aa•||1$•core)
   (
      echo 'chl>Brror:c/hl>Thi•  page wa• called incorrectly';
      exit1

   el••
   (
      $date•  date(  'r  d,  Y'  );                    '''

      // cree un documento pdfen  -moria
      $pdf • pdf_new();
      pdf_opea_fl1e1$pdf1;

      // defina el nombre de la fuente que••  utilizar& po•teriormente
      $fontaa•• • 'Time•-R — n' 1

      // defina el tamafto da la p&gina y cr6ela
      il la carta OS•• 11• x 8.5"  y hay aproximada—ate 72 punto• por pulgada
      $width • 11*721
      $height • 8.5•721
      pdf_begin_page($pdf,  $width,  $height)1

      // dibuje lo• borda•
      $inHt  • 201  // eQ&cio entre el  borda y el  borde da la p&gina
```



```
$bordar•  101  // aac:hura da la Unn  del borda principal
$ianer • 21  // ••pacio dentro del borda

//dibuje el borda exterior
pdf_rectl$114f,  $iuet-$iDMr,
          $in••t-$iuer,
          $width-2*(Siaaet-$ianerl,
          $119ight-2•($ia..t-Si11119rl11
pdf_•troke($pdfl1

//dibuje el borde principal con -   uc:hura da $bordar punta
pdf_••tlinewidth($fdf,  $borderl1
pdf_rect($pdf,  $iuet+$bordar/2,
          $iaaet+$bordar/2,
          $wldth-2•($in•et+$boinler/21,
          Sheight-2•($ia8et+$110rder/211,
pdf_•troke($pdf)1
pdf_•etliaewidth1$pdf,  1.011

// dibuje el borda interior
pdf_rect($pdf,  $in•et+$bordar+$inaer,
          $in•et+$bordar+$ianer,
          $width-2*($in••t+$border+$iaaer),
          $beight-2•C$ineet+$border+Sillnerl11
pdf_•troke($pdf)1

// afteda el titulo
Sfont • pdf_fiadfontl$pdf,  $fon~.    'ho•t',   01,
if  f$fontl
   pdf_•etfont($pdf,  $font,  4811
S•tartx • ($width - pdf_•tringwidth($pdf,  'PIIP Certification',
            $font,  4811 /21
pdf_•how_xyl$pdf,  'PIIP certificetion',  S•tartx,  4901;

/l afteda texto
$foat • pdf_filldfont($pdf,  $foat —.  'ho•t',   0);
if  ($fontl
   pdf_..tfoatC$pdf;  $font,  2'11
$•tartx • 701
pdf •how xy($pdf,  'Thi• 1• to certify that:',   S•tartx,  430);
pdf=•how:xyl$pdf,  •trtoupper1$-I,  $'•tartx+to,  39111

$font • pdf_fillllfontl$pdf,  $fon —.  'hoet',   0)1
if  1$fontl
   pdf_..tfont($pdf,  $font,  2011

'ha•  4 —•tnted  that they are certifiable  '.
•by pa••iag a rigoEOU• eu••,  S•tartx,  34011
pdf_•how_xy($pdf,  •coaei•tlag of tb.rff -lt1ple  choice que•tion•.'
                $etartx,  31011

pdf_•b.ow_xyl$pdf,  '•—   obtaiaed a •con  of S•con•.  •t.',   $etertx,  26011

pdf_•how_xy(Spdf,  •The te•t wa• ••t aad over•••n by th•  •,  S•tartx,  2101;
pdf_•how_xy($pdf,  •rictiaaal Iaatitut• of  PIIP Certification',
            S•tartx,  1101,
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
pcll. .a.. ayCtplf,  ·- $4aCe.•,  tacasu,  15011
lNK_aTlo1t_ay1Spc1f,  'Allcbariaed by,•,  sacartz,  1001,

// dada  la  l...-  de .... de bica de la f1-
// ~      qu. e....  11119 c:aalliar -la nea al arcla:Lw ele la Una
Spacb • •C,/Pngr  ..  l'Uaa/Apac:ba  Gnllp/Apacllll/llcdaca/pllfmraqUa/
c:bapurll/' ,

// al uN·de  la ftn:la. 9:Lf para PDl'Lib aa WiJIIIGll8.... 1. s-rar prQbl —•
taisaac11Z9 •, p00_opaa_illaga_filaCSplf,  '9:Lf•,  s,at.1a.'a1saacura.9if',
                                  01,
PIK...J>laca_iusaCSpdf,  S•:Lsaacura,  200,  75,  111
pdf_clo.. _iusaltpdf,  Sai911&Cural,

// def:Laa  1- col-••  para al aallo
pdf_Mccolor(Spdf,  'f:Lll',  'rsb',  0,  0,  .••  01,  // anl  oac:uro
pdf_aall:color(Spdf,  •acroli:a',  'EP',  0,  0,  0,  OI,  //  negro

// dibuja  la c:l.aca  1
pdf_—co(Spdf,  no.  1501;
pdf_l:LaacoCSpdf,  ,10,  5511
s,dt_l:l.aall:o(Spdf,  ,,2.  11,
pdf_l:LaacoCSpdf,  ,.,.  ,,>,
pdt_liaaco(Spdf,  ' "·  1501;
pdf_cloaepatJaCSpdfl ·
pdf;..fill(Spdf);

// coall:orao da la cinca  1
pdt_—coCSpdf,  no.  1501 i
pdf_l:1.aaCo(Spdf,  ,10,  5511
pdt~l:LnacoCSpdf,  ,,2,  ,,,,
pdf_l:1.aaCo(Spdf,  tt,,  t911
pdf_uaaco(Spdt;  ' "·  1501,
pdt_cloaapacb(Spdfl :
pdf_acroli:a(Spdtl,

// dibuja  la ciaca  2
pdf_—11:oCSpdt,  ,,0,  15011
pdt_lina11:o(Spdf,  ,10,  ,,>,
pdf_liaall:o(Spdf,  ,1s,  ,111
pdf_l:Lnaco(Spdf,  711,  5511
pdt_liaaco(Spdf,  ,1,,  1~01,
pdt_cloaapacb(Spdfl1
pdf_f:Lll(Spdtl ·

// coacoaio de la ciaca 2
pdf_.....coCtplf,  ff0,  15011
pdf_l:LaacoCSpdf,  ,10,  ti);
pdf_Uilall:o(Spdf,  HS,  HI ·
pdf_linaco(Spdf,  11,,  551;
s,dt_linaco(Spdf,  ,,,,  15011
pdf_clONpaCIL(Spdfl ;
pdt_acroli:aCSpdf>,
pdf_Mt¡color(Spdt,  •fill•,  'EP',  .1,  o.  o,  o>,  //  cojo

//dJ:t,a.ja al ••llo
```

```
••—•e&rC"'·  115,  ,2,  5'7,  H,  SPdt,  cr.e>,

//concomo  del  ..110
drav_acar(H5,  ns,  32,  57,  10,  Spdf,  ra1.. 1,

//  cffline la pAsiaa y pnpu9  -  ~
pdf_and_pqa(tpdfl ·
pdf_clOH(~)  ·
Sdaca • pdf_sac_buffarCtplfl,

//  gaara  loa ancabaaadoa pua  que al na.....-r  NIeacima
// la apUcaci•  cnncca
baader('cancaac-cypa,  appUca11:ioil/pdf '1 ·
baadar(•cancaac-diapoaician,  inlinal  fil......C..C.pdf'~,
baaderc'cancanc-1anscl1,  ' '  acrleaCSdacall;

// nprueca  al  l'DI'
ac:bo  Sdaca,


f11DCC1-  drav acarCSceall:an,  Sceall:ny,  Spo:Laea,  Srmd:Lu,
              -      Spoinc_aise,  Spdf,  Sfilledl

  Sinner_radiua  •  Sradiua-Spoin11:_aiser

  for  CSi  •  0,  Sic•tpoiaca•2,  Si++  1
  (
     Sangla-  111•2~11111 (Spoiaca•21 r

     if(Sit21
     (
        sx  •  Sradiua•caa CS...lal  •  Scencan,
        Sy  •  Sradiua•aiaCSaaglal  •  Sceacnyr

     alM
     {
        Sx  •  $:Lanar radiua•coa(SaDll•I  + SceaC•n,
        ty  •  S:Lanar:r--siua•ainCSantl•I  •  Sceacny,
     }
     f(S:L•  •01
        pdf_-co(tpdf,      Sx.  t'/1:
     al••  if(S:L••Spoinca•21
        pdf_cloaapatJa(Spdfl:
     al••
        pdf_liaaco(tpdf,  Sx,  Syl:
  }
  u (Sf:Llledl
     pdf_f:Lll_acl00ll:a(Spdfl:
  .1 ..
     pdt_acroli:e(tpdfl;
```

El cmificacloque se obtiene al utilizar esta secaencia de CIIIIMMCl8 se mlMSlra en la figura 32-7. Como puede comprobar, es .,_.. .. similar a los otrm, a excep-

ción c¡.e que el borde es m4s sencillo y la estrella tiene un upecto diferent& Se debe a que los hemos dibujado en el documento en lugar de utiliza un archivo existente.

Flgu111 32.7. La secuencia de comandos pdftlb.php dibuja el cei'tlflcado en un documento PDF.

Analizaremos las partes de esta secuencia de comandos que son diferentes a las de los ejemplos anteriores.

Los visitantes tienen que obtener sus datos en un certificado, por lo que crearemos el documento en la memoria en lugar de un archivo. Si lo escribiéramos en un archivo, tendríamos que preocupamos por mecanilffhospara ~ar nombres de archivo exclusivos, para evitar que se vean los certificados de otros usuarios y para determinar la forma de eliminar los archivos antiguos para liberar espacio en el servidor. Para crear un documento en memoria, invocamos pdf_new () sin parámetros seguido de una llamada a pdf_open_file (), como se indica a continuación:

```
Spdf • pdf new(l1
pdf_open_file($pdf, ''l1
```

Nuestro borde simplificado estará formado por tres barru: un borde grueso y dos más estrechos, uno dentro del borde principal y otro exterior. Los dibujaremos como rectángulos.

Para situar los bordes de forma que se pueda modificar el tamafto de la página o el aspecto de los bordes, basaremos todu las posiciones de los mismos en lu

variables que ya ll!nemal, $width y $height, y en algunas nuevas: $inaet, $border y $inner. Ulilizalemos $inaet para especificar la anchura en puntos del borde con respecto al borde de la página, $border para especificar el grosor del borde principal y $inner para especificar la anchura del espacio entre el borde principal y los más esln!Chos,

Si ha dibujado con otro API gráfico, el uso de PDFlib le deparará algunas sorpresas. Si no ha lefdo uno de los capftulos anteriores sobre generación de ~genes, deberla hacerlo ahora ya que la generación de imágenes a,n la bibliotecagd es muy similar al uso de PDFlibpara lo mismo.

Los bordes estrechos son sencillos. Para crear un rectángulo, utilizamos pdf_rect (), que requiere como parámetros el identificador del documento POP, la coordenada x e y de la esquina inferior izquierda del rectángulo y la anchura Y la altura del rectángulo. Como queremos que el disefto sea flexible, calcularemos estos valores a partir de las variables que hemos definido:

```
pdf_rect(Spdf, $ln••t-$lnner,
               $inHt ·$imser,
               $vidtlt.·2•(Sln••t-$inner),
               $heigllt-2•($l119et-Sinnerll1
```

La invocación de pdf_rect () establece un trazado en fo~ile rectángulo. Para poder dibujar esta forma, tendremos que invocar la función ~f_stroke (l, como se indica a a,ntinuación:

```
pdf_•troke"pdfl1
```

Para poder dibujar el borde principal, tendremos que especificar la anchura de las líneas, que de forma predeterminada es de 1 punto. La sigillfflte llamada a pdf_setlinewidth C) la establece en puntos $border (en este caso 10):

```
pdf_Htlinewldth($pdf, $borderl1
```

Una vez definida la anchura, creamos de nuevo un rectángulo con pdf_rect (l e invocamos pdf_stroke () para dibujarlo.

```
pdf_rect1$pdf, $in..t+Sborder/2,
               $1119et+$border/2,
               Svidth·2•1Sln••t+$border/21,
               $heigllt·2•1$l119et+$border/21l;
pdf_•troke($pdfl1
```

Oespu'5 de trazar la línea de la anchura, tendremos que volver a establecerla en 1 con el siguiente código:

```
pdf_•etlinevidth($pdf, 1.01l
```

Utilizaremos pdf, show xy ( ) para situar cada línea de texto en el certificado. En la mayorfa de las Ífneas de texto, empleamos un margen izquierdo que se puede configurar ($atartx) a,mo coordenada x y un valor seleccionado a 0;0 como coordenada y. Como queremos que el título aparezca centrado en la página, tendremos

EL_HOLISTICO@HOTMAIL.COM - 2006

que conocer su anchura para situar la parte izquierda del mismo. Para obtener la anchura, utilizamos pdf_stringwidht (). La llamada

```
pdf_stringvidthl$pdf,    'PHP Certification'l;
```

devuelve la anchura de la cadena 'PHP Certific:ation' a,n la fuente y el tamaño de fuente actuales.

Al igual que con las otras versiones del certificado, incluiremos una firma como mapa de bits escaneado. Las tres siguientes instrucciones

```
$signature  • pdf_open_i—ge_file  ($pdf,  'gif ',  $path.  'dgnature.gif  ',
                                  -—k',   0l;
pdf_place_i  ..ge($pdf,  $signature,   200,  75,  11;
pdf_clo••~l—geC$pdf,    $Si9:"&turel;
```

abren un archivo GIF que contiene la firma, añaden la imagen a la página en la ubicación especificada y cierran el archivo GIF. Tambictn se pueden utilizar otros tipos de archivo. El único parámetro que quizás no este! demasiado claro es el quinto parámetro de pdf_plac:e_image(). Esta función no se limita a añadir la imagen con su tamaño original. El quinto parámetro es un factor de escala. Hemos optado por mostrar la imagen a tamaño real y hemos utilizado 1 como factor de escala, pero podríamos haber utilizado un valor mayor para aumentarla o una fracción para reducirla.

E(elemento más difícil de añadir al certificado es el sello. No podemos abrir automáticamente y añadir un metarchivo de Windowsq:uecontenga el sello que ya tenemos, pero podemos trazar cualquier forma.

Para trazar una forma con relleno como una de las cintas, podemos escribir el siguiente código.

En este caso definimos el trazo o el color de la línea en negro y el relleno o color interior en azul marino:

```
pdf_setcolor($pdf,    'fill',   •rgb',  0,  o,  .4,  01;  // azul  oscuro
pdf_setcolorl$pdf,   'stroke',   'rgb',  o,  o,  o,  01,    // negro
```

A continuación definimos un polígono de cinco lados como una de las cintas y lo rellenamos:

```
pdf_lllOVetol$pdf,   630,  1501;
pdf_linetol$pdf,   610,  551;
pdf_linetol$pdf,   632,  691;
pdf_lineto($pdf,   646,  491;
pdf_liaatol$pdf,   666,  1501;
pdf_closepath($pdfl;
pdf_fill  1$pdf);
```

Como tambictn queremos que el polígono tenga un contorr.o, tendremos que definir el mismo trazado por segunda vez, pero invocaremos pdf_stroke () en lugar de pdf_fill ().

Como la estrella de varias puntas es una forma compleja y repetitiva, hemos esaito una función que calcula por nosotros las posiciones del trazado. La función

es draw_star () y requiere coordenadas x e y para el centro, el número de puntos necesarios, el radio, la longitud de los puntos, un identificador de documento PDF y un valor booleano para indicar si la forma de estrella tendrá relleno o solamente uncontomo.

La función draw_star() utiliza conceptos de trigonometría básicos para calcular la posición de una serie de puntos para trazar nuestra estrella. En cada punto de la estrella, buscamos un punto del radio de la estrella y un punto en un circulo más pequei\o $point _si ze dentro del círculo interior, y trazamos una línea entre ambos. Conviene recordar que las funciona de trigonometría de PHP como c:os !) y sin () trabajan con radianes y no con grados.

Por medio de una función y de algunos cálculos matemáticos podemos generar una forma compleja y repetitiva con gran precisión. Si hubictramos necesit~d~ un modelo de borde más complicado, podríamos haber empleado un enfoque similar.

Una vez generados todos los elementos de la página, tendremos que finalizar la página y el documento.

Solucionar problemas con los encabezados

Un aspecto que conviene mencionar en todas estas ~encías de co~ndos es que tendremos que indicar al navegador que! tipo de datos vamos a enviar. Para ello enviamos un encabezado HTTP de tipo de contenido, como por ejemplo

```
header(  •eontent-type:   application/msvord'   l;
```

o

```
headerl  •content ·type:   application/pdf'   l;
```

Debe saber que los navegadores procesan estos encabezados de forma poco coherente. En concreto, Internet Explorer suele ignorar el tipo MIME e intenta detectar automáticamente el tipo de archivo, aunque parece que en las últimas versiones se ha mejorado este problema. Algunos de nuestros encabezados pueden resultar problemáticos con encabezados de control de sesión, para lo que existen distintas soluciones. Una de ellas consiste en utilizar parámetros GBT en lugar de POST o parámetros de variable de sesión. Otra solución es no utilizar PDF en línea sino que el usuario lo descargue, como indicamos en el ejemplo Hello World.

Tambictn se pueden evitar muchos de estos problemas si se escriben dos versiones del código düerentes, una para Netscape y otra para Internet Explorer.

Ampliar el proyecto

La inclusión de tareas de evaluación más realistas mejoraría sin duda este proyecto, pero la intención de nuestro ejemplo es mostrar distintas tknicas de enviar

EL_HOLISTICO@HOTMAIL.COM - 2006

documentos. Entre los documentos personalizados que puede que le interese enviar en línea se pueden incluir documentos legales, formularios de pedido o solicitudes parcialmente completados o formularios para organismos gubernamentales.

Lecturas adicionales

Le sugerimos que visite el sitio de Adobe si necesita información adicional sobre los formatos PDF (y FDF), http://www. adobe. com.

A continuación

En el siguiente capítulo analizaremos las nuevu funciones XML de PHP 5 y utilizaremos PHP para conectamos al API de servicios Web de Amazon por medio de RESTy SOAP.

EL_HOLISTICO@HOTMAIL.COM - 2006

33

Your feedback

$toaddress = 'informacion@ejemplo.com';

con XML y SOAP

En los últimos al\os, XML (lenguaje de marcado extensible) se ha convertido en uno de los medios de comunicación más importantes. En este capítulo utilizaremos la nueva interfaz de servicios Web de Amazon para crear un carro de la compra en nuestro sitio Web local que utilice Amazon como sistema (llamaremos a esta aplicación Tahuayo, nombre de un afluente del Amazonas). Para ello utilizaremos dos métodos diferentes: SOAP y REST. REST también se conoce como XML sobre HTIP. Emplearemos la biblioteca interna SimpleXML de PHP y la biblioteca externa para implementar ambos métodos.

Nos centraremos en los siguientes aspectos:

- Fundamentos de XML y SOAP
- Utilizar XML para comunicamos con Amazon
- Analizar XML con la biblioteca SimpleXML de PHP
- Almacenar en caché las respuestas
- Comunicarse con Amazon a través de NuSOAP

El problema

En este proyecto tenemos dos objetivos. El primero de ellos consiste en comprender qué son XML y SOAP y cómo utilizarlos en PHP. El segundo nos permitirá

EL_HOLISTICO@HOTMAIL.COM - 2006

utilizar estas tecnologías pua comunicamos con el mundo exterior. Hemos seleccionado el programa Servicios Web de Amazon como ejemplo que puede resultarnos dtil pua nuestro sitio Web.

Amazon ofrece desde hace tiempo un programa asociado que le permite anunciar productos de Amazon en su propio sitio Web. Los usuuios pueden seguir un enlace hasta la p4gma de cada producto en el sitio de Amazon. Si pulsan sobre estos enlaces desde nuestro sitio y adquieren dicho producto, recibiremos una pequefta comisión.

El programa de Servicios Web nos permite utilizar Amazon como si fuera un motor: se puede buscar en el mismo y mostrar los resultados desde nuestro propio sitio, o se puede llenar directamente el carro del usuario con los artfculos que haya seleccionado mientru navegaba por nuestro sitio. En otru palabras, el diente utiliza nuestro sitio hasta que quiera puu por caja, operación que debe realizar a trav._ de Amazon.

Las comunicaciones entre nosotros y Amazon se pueden establecer de dos formu diferentes. Por un lado, por medio de XML sobre HTI'P, lo que también recibe el nombre de Transferencia de estado de representación (REST en ingi..). Si por ejemplo queremos realizar una bdsqueda con ayuda de este método, enviamos una solldtud HTTP normal de la información que necesitemos y Amazon responder4 con un documento XML en el que se incluya la información solicitada. Tru ello, se puede analizar este documento XML y mostrar los resultados de la bdsqueda al usuario final. El proceso de envio y recepción de datos a trav._ de HTI'P es muy sencillo pero la facilldad de analizar el documento resultante depende de la complejidad de &te. La segunda forma consiste en utilizar SOAP, uno de los protocolos est4ndar de servicios Web. Anteriormente equivalfa a Protocolo de acceso a objetos simple, pero recientemente se decidió que dicho protocolo no era tan simple y que el nombre resultaba ligeramente confuso. Como resultado, el protocolo se sigue llamando SOAP, pero ya no es una sigla.

Generaremosun cliente SOAP que pueda enviar solicitudes a Amazon y recibir lu correspondientes respuestu del senridor SOAP de Amazon. Tendl'Úl la misma información que lu respuestas obtenidas por medio.delIMtodo XMLsobre HTTP, pero utilizaremos un enfoque diferente para extraer los datos, b4sicamente la biblioteca NuSOAP de PHP.

Nuestro objetivo final en este proyecto es crear nuestro propio sitio Web de venta de libros que utilice Amazon como sistema. Creuemos dos versiones alternativu: una que utilice RESTy otra que utilice SOAP.

Entender XML

Nos detendremos brevemente para analizar XML y los servidos Web, en caso de que no estf familiarizado con estos conceptos.

XML es el Lenguaje de marcado extensible. La especificaci6n est4 disponible en el W3C. También puede encontrar numerosa información sobre XML en el sitio XML de W3C, en la dirección http://www.wl. org/XML.

XML se derin de SGML (Lenguaje de marcado gmeralizado eldndar). Si ya c:cnoce HTML (y suponemos que llegados a este punto del libro, sin duda lo hace), no le resultar6 diflcil enlmder los concepl0B de XML

XML es un fonnato de texto para doc:umenlOB basado en etiqueta Como ejemplc:ide documento XML,.111 el listado 33.1 se incluye una delu respuestas que envfa Ámazon como contestación a una solicitud XML sobre HTI'P.


```
c?:icml veraiODoo"1.0" aiaaocliD!l-"ln'l'·I"?>
cProductlDfo JIIIID$1..i•"bttp,//www.w3.org/2001/111L1c:11811a-iaac.ace•
x•i1Dolr...•pac~tiOD          •"bttp,//Jllll .... soe.coa/ac:b....-2/dev·
beavy.xad">
      cDetail• 11rl•"bttp://www .... soe.C011/exec:/obidoa/redirect?
         tag•taagl~•i91112&creative••lllllllllll•ll2&
         0~ ·202s12,11llt_c:odlt•Xl1212&path-AII•/o&723171t2">
         cAlliD>0,723171t2c/AlliD>
         cProducCS->,.P   alld llySQL Web DevalopMDtc/ProducUS->
         cCatalog>II00ltc/Catalog>
         cAutbon>
            cAutbor>Luke Wellill!lc/Autbor>
            cAutbor>La..ra Tllomamlc/Author>
         c/Autbora>
         ca.1..aeDate>lO Naroh, 2001c/a.leaaaDate>
         cllanufacturer..s...c/Namafacturar>
         clugeurlS-11>
            http://image• .... soe.coa/iugea/P/0172317142.01.TIIUIIBlll.jpg
         c/I.ageurl...11>
         clugeurlllllldi->
            http!//imagea .... s011.C011/1..ge•/P/06723171t2.0Í.lllalll&l.jpg
         c/I.age!Jrlllllldi-
         clugeurlLarge>
            http://i.agea ....sOD.COll/i..ge•/P/0672317142.01.Llllllll.jpg
         c/IugeUrlLarge>
         cLtatPrice>$41.99c/Li•tPrice>
         cOUrPrioa>$lt.91c/08rPrice>
         cU.edPrioa>$11.ISc/UeeclPrice>
         cThirdPartyllewPrice>$11.75c/ThirdPartyllewPrice>
         c8ale•llmlk>112c/8ala•llallk>
         cLiata>
            cLi•tld>lKllflIVIQIIIISFc/Li•tld>
            cLi•tld>22YC01IGPIIJlc/Li•tld>
            cLi•tld>Y219Bl,2QXVXc/Li•tld>
         e/Lista>
         cBrowaeLi•t>
            clrowaellode>
               clrowH•—>PRP (Computar progr.. languagac/Browae.._>
            c/IrowaellOcle>
            clrowaellOcle>
               clrow......>IOL (COIIPUter progr•• l1111g11agac/lrow••W..>
            c/lrowHIIOde>
            clrowaellOcle>
               clrowa••...>Web aitaac/lrow••••..>
            c/lrowH•ode>
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
        clr—Hllocle>
          car-•ell...>Deaignc/lrow..Hame>
        c/Brow•ellode>
        car-•dode>
          carowad...>SQL (C:ompucer l&IMJU&gel c/lrowaeJlaM>
        c/Browaellocle>
        clrowHllocle>
          carowaell&M>Sql (Progra..ing Languagelc/lrow•••ame>
        c/lr-••llode>
        clr-edode>
          carowadame..C:ompucer ••cworkac/ar-.e•....
        c/lrowadode>
        clrowadode>
          clrowad...>C:ompucer lk• - Languagea / Programu.ngc/lr-•eJI••>
        c/Br-aellode>
        car-.eHode>
          carowaell&M>C:o.pucerac/lr-.d ...>
        c/Br-••Hode>
        c8rowHHode>
          c8row...AM>Progra..lag Language• - Generalc/Brow•••AM>
        c/arowadode>
        clrowHllocle>
          car-•eHAM>Incernec - Web Sice Deaignc/lrow•••.....
        c/lrow•••ode>
        clrowaellocle>
          clrowad...>D&Caba•• Managemenc - SOL Serverc/lrowae11.....
        c/ar-.ellode>
        clr-•ellode>
          clr_..._ ..,EO!lr-1av Languagea - SQLc/ar-.....
        c/Browaellode>
      c/lrowHLiaC>
      cNedia>Paperbackc/Nedia>
      cllmfledia>lc/lfu•M•cUa..
      ciabn>067231784Zc/Iabn>
      cAvailabiliCy>Uaually ahipa wichin 24 hourac/AvailabiliCY>
      cSi•ilarProducCa>
        cProduCC>0735709Zllc/Product>
        cProdUCC>18&100J730c/Product>
        cProduCC>073570970Xc/ProduCt>
        cProduCC>186100&918c/Product>
        cProduCC>059&000413c/Product>
      c/SimilarProducta>
    c/Decaila>
  c/Prodl&CCinfo>
```

El documento comieaza con la siguiente linea:

Es una declaración estai.ndar que nos indica que el siguiente documento ser4 XMLy utilizará la codificación de caracteres UTF-8.

Fijese en el c:uerpo del documento. Está. enteramente formado por pares de etiquetas de apertura y de ci~ como por ejemplo:

```
cProduct• — >PRP and NySQL lfeb Developmencc/Prodl&CC — >
```


ProcluctName es un elemento, al igual que lo seria eri HTML Y, al igual que en HTML, podemos anidar elementos:

```
cAucbora>
   cAuchor>Luke Wellingc/Author>
   cAutbor>Laura Thomaonc/Author>
c/Auchon>
```

Como suc:ede tambim en HTML, los elementos pueden tener atributos:

```
cDecaila url•"bCCp://www.a ..zon.call/exec/obidoa/redirecc?tag•
tangledwabdeaignlZ&cnacive•XXX•iii••X•••IIZ'c:allp•202Sl2&link_code
•Xl21Z&path•ASIH/0672Jl784Z">
```

Este elemento Detai 1s tiene un único atributourl. Como el URL es extenso, se ha dividido en varias líneas de código.

También existen diferencias con respecto a tlTML En primer lugar, todas las etiquetas de apertura deben tener una etiqueta de derre. La excepdón a esta regla son los elementos vados que se abren y derran en una misma etiqueta, ya que no encierran texto alguno. Si está. familiarizado con XHTMLhabr4 comprobado que se utiliza la etiqueta cbr />en lugar de cbr> por esta razón.

Adem6s, todos los elementos deben anidarse correctamente. Un analizador HTML podría pasar por alto Texto pero para que se considere código XML o XHTML válido, las etiquetas tendrfan que anidarse correctamente: Texto.

La prindpal diferencia entre XMLy HTMLes que da la sensación de que creamos nuestras propias etiqueto sobre la marcha. Se debe a la flexibilidad de XML. Podemos estructurar nuestl'Os documentos para que coincidan con los datos que queremos almacenar. Podemos formalizar la estructura de los.documentos XML si esaibimos una OTO (Definición de tipo de documento) o un Esquema XML Ambos documentos se utilizari para desaibir la estructura de un determinado documento XML Si lo prefiere, diremos que la OTOo el Esquema son como la declaración de una clase *y* que el documento XML es como una instanda de dicha clase. En nuestro ejemplo no utilizaremos OTO ni Esquemas.

Puede leer la OTO de Amazon para este documento en la siguiente dirección http: //xml. amazon.c09./ache .. a2/dev-beavy .dtd. Puede leer el Esquema XMLen http: / / xml • amazon. com/ achemas2/dev-heavy. xad.

En algunos navegadores no podr4 abrir el archivo OTO, ya que tratan de analizar la OTOcomo XML *y* se producen confusiones. Sin embargo, puede descargarlo y leerlo en el editor que prefiera. El Esquema XML deberla poder abrirlo directamente en su navegador.

Habr4 comprobado que, aparte de la declaración XMLinicial, la totalidad del c:uerpo del documento se incluye dentro del elemento Productinfo. Es lo que se deriodüna elémento ral% del documento. Vdmoelo con mú detalle:

```
cProcluctinfo x.1Da,aa1.•bctp://www.w3.org/2001/XMLSchema-inacaace•
     xai:noll.-..paceScllem&Locacion••bccp://xml.... sOA.C011/acb...•z/
     dev-llN.vy.ud">
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Comprobará que tiene algunos elementos poco habituales. Se denominan Espacios de nombres XML. Para los objetivos de este proyecto no·es necesaiio que comprenda los espacios de nombres, pero pueden resultar muy útiles. La idea búica es calificar nombres de elementos y atributos con un espacio de nombres para que los nombres comunes no produzcan eñores al utilizar documentos de diferentes fuentes.

Si necesita más información sobre espacios de nombres, puede leer el documento "Namespaces in XML Recommendations·,en la dirección http://www. w3. org/TR/RBC-xml-namea.

Si quiere conocer mú sobre XML,existe una amplia variedad de fuentes..El sitio de W3C es un excelente punto de partida y, literalmente, existen cientos de libros y tutoriales Web. ZVON.org es uno de los mejores basado en la Web.

Servicios Web

Los servicios Web son interfaces de aplicación que se ofrecen a través de la World Wide Web. Puede considerar un servicio Web como una clase que expone sus métodos públicos a través de la Web. Los servicios Web se han hecho muy populares y algunos de los principales nombres del sector empresarial ofrecen su funcionalidad a través de este tipo de servicios. Por ejemplo, Google, Amazon, eBay y PayPal ofrecen en la actualidad una gran variedad de servicios Web. Después de que en este capítulo definamos un cliente a la interfaz de Amazon, comprobará que la creación de una interfaz cliente a Google resulta muy sencilla. Encontrará ~ información al respecto en http: //www.google.com/apia.

Una lista cada vez mú amplia de servicios Web se encuentra disponible en http://www.xmethoda.net.

Existen diversos protocolos básicos implicados en esta metodología de invocación de funciones remotas. Los dos mú importantes son SOAP y WSDL.

SOAP

SOAP es un protocolo de mensajería dirigido por solicitudes y respuestas que permite a los clientes invocar servicios Web y permite a los servidores responder. Todos los mensajes SOAP, ya sean solicitudes o respuestas, son simples documentos XML:En el listado 33.2 se recoge un ejemplo de las solicitudes SOAP que podemos enviar a Amazon.

Llst.do 33.2. Solicltud SOAP de una búsqueda basada en ASIN.

```
c?xml ver•ion.•1.0• eacocling·•~r~a·,~
cSOAP·ERV:Bnvelope  xalae:SOAP·IIIV•"bttp,//ec:lle9ee.xal80ap.org/eoap/
eavelope/•
        xmla•:SOAP- ..C•"bttp,//ec:lle9e•.xal•oap.org/•oap/eacoc:liag/•
        xmla•,x•i•"bttp://www.wl.org/2001/XNLScbella-iaetaace•
        xmla•:x•d••bttp:~/www.wl.org/2001/XNLScb ...•
```



```
   SOAP-IIIV,eacodiagStyl••"bttp://ecbellae.xaleoap.org/eoap/eacadiag/">
   cSOAP·IIIV\10dy>
      cllU18$¡ll:Aeinsearcblteque•txalae: .....epl•°uzn:PI/DeVCentral/SoapService•>
         cAeinSearcbllequeet xei:type•••!Aeialleque•t">
            caeia >00&0511057c/a•in>
            eta~ >yoll~····ociate-idc/tag>
            ctype >beavyc/type>
            cdev-tag >your•deY•tagc/dev·tag>
         ·c/AeiaSearcbllequeet>
      c/.....spl:AeinSearcbllequeat>
   c/SOAP•IIIVtlodY>
c/SOAP•IIIV1lavelope>
```

El mensaje SOAP comienza con la declaración de que se trata de un documento XML El elemento raíz de todos los mensajes SOAP es el sobre SOAP. Dentro de éste se incluye el elemento Body, que contiene la verdadera solicitud.

Esta solicitud es una AainSearchRequeat, que pide al servidor Amazon que busque un determinado artículo en la base de datos en función del ASIN, el número de artículo estútdar de Amazon.com. Se trata de un identificador exclusivo asignado a cada uno de los productos de la base de datos de Amazon.com.

AainSearchRequeat es una especie de llamada de función en un equipo remoto y los elementos que contiene este elemento podrían ser los parámetros • se pasan a dicha función. En este caso, pasamos un ASIN para el libro de Dilbect *WtlJ of tM Wtt1MI.* También será necesario pasar la etiqueta, que es su Id. de ~ ,el tipo de búsqueda que se va a realizar (heavy o 1 i te) y el dev-tag, el valor que le asigna. Amazon com. El tipo de elemento indica al servicio si queremos información general (1 i te) o toda la información disponible (heavy).

La respuesta a esta solicitud es muy similar al documento XML que vimoe en el listado 33.1, pero se encierra en un sobre SOAP.

Al trabajar con SOAP, tendremos que generar solicitudes SOAP e interpretar respuestas por medio de programación con ayuda de una biblioteca SOAP, independientemente del lenguaje de programación que utilicemos. Es muy positivo ya que nos evita tener que generar la solicitud SOAP e interpretar la respuesta manualmente.

WSDL

WSDL significa Lenguaje de descripción de servicios Web. Se emplea par~ describir la interfaz a los servicios disponibles de un determinado sitio Web. Si quiere ver el documento WSDL que describe los servicios Web de Amazon que utilizaremos en este capitulo, lo encontrará en la dirección http: //aoap.-amazon.com/•chemaa2/AmazonWebServicea.wadl.

Como comprobará si utiliza este enlace, los documentos WSDL son considerablemente mú complejos que los mensajes SOAP. Si tiene la oportunidad, tendrá que generarlos e interpretarlos mediante programación.

Si quiere saber mú sobre WSDL. puede consultar el proyecto de W3C en la dirección http: //www: w3. org/TR/—dl20.

EL_HOLISTICO@HOTMAIL.COM - 2006

Al cierre de la edición de este libro, WSDL no era todavía una Recomendación W3C por lo que es posible que se modifique. Sin embargo esto no ha impedido a los programadores que lo utilicen con gran entusiasmo. Pero al igual que todas las piezas del puzzle que componen los servicios Web, siempre puede cambiar debido a su novedad.

Componentes de la solución

Para generar nuestra solución, tendremos que detenemos en diferentes aspectos. Además de los elementos más e,identes, como la interfaz de carro de la compra que mostrar a los clientes y el código para conectarse a Amazon, se necesitan otros componentes.

Tras recuperar un documento XML, el código debe analizarlo para extraer la información que mostranl el carro. Para cumplir los requisitos de Amazon y mejorar el rendimiento, tendremos que tener en cuenta el almacenamiento en caché. Por último, como el paso por caja tendri que realizarse en Amazon, necesitamos algún tipo de funcio~lidad para pasar los contenidos del carro del usuario a Amazon y pasar al usuario a dicho servicio.

Crear un carro de la compra

Ser6 necesalio generar un carro de la compra como interfaz de nuestro sistema, algo que ya hem~ hecho en un capítulo anterior. Como los carros de la compra no son la parte prlnapal de este proyecto, utilizaremos una aplicación simplificada. Basta con proporcionar un carro básico para poder realizar el seguimiento de lo que quiere comprar el cliente e informar de ello a Amazon en el momento de efectuar la compra.

Utilizar las interfaces de servicios Web de Amazon

Para utilizar la interfaz de servidos Web de Amazon, tendnl que descargar el paquete Amazon Web Services Developers Kit, de http://www. amazo::l. com/gp/aw•/landing .html.

Tambifn tendr6 que solicitar, en el mismo sitio, un Developer Token. Se utilizanl para identificar las solicitudes que remita a Amazon.

Tambifn puede obtener un Id. de asociado a Amazon, que le permitirá recibir una comisión si los usuarios realizan compras a través de su interfaz.

Una vez descargado el paquete, léalo. Incluye documentación sobre el ñmcionamiento de la interfaz y ejemplos de código en distintos 'lerigwqéj, incluyendo PHP.

Antes de poder descargarlo, tendñ que aceptar el contrato de licencia. Conviene que lo lea atentamente, ya que no se trata del típico contrato de licencia. Algunas de las condiciones importantes durante la implementación son las siguientes:

- No debe realizar más de una solicitud por segundo.
- Debe almacenar en caché los datos que provengan de Amazon.
- Puede debe almacenar en caché datos durante 24' horas y algunos atributos estables durante un máximo de tres meses.
- Si almacena en caché los precios y la disponibilidad durante más de una hora, tendnl que indicarlo en la p6gina.
- Debe vincular todos los datos de Amazon a la página de Amazon y no puede vincular texto o gráficos descargados de Amazon a otros sitios Web comerciales.

Con un nombre de dominio dificil de pronunciar, sin promoción y sin una razón aparente para utilizar Tahuayo.com en lugar de acceder directamente a Amazon.com, no tendremos que hacer nada más para conseguir menos de una solicitud por segundo. Hemos implementado el almacenamiento en caché para cumplir las condiciones expuestas anteriormente. Almacenamos en caché las imágenes durante 24 horas y los datos de los productos (incluyendo los precios) durante una hora.

Hemos optado por ignorar el punto *S*. Queremos que los artículos de la página principal estfn vinculados a p6ginas de nuestro sitio y que solamente se vinculen a Amazon cuando estfn completos.

Analizar XML

La primera interfaz que o&ece Amazon a sus servicios Web es a través de REST. Esta interfaz acepta una solicitud HTIP normal y devuelve un documento XML Para utilizar esta interfaz tendremos que analizar la respuesta XML que nos envíe Amazon."Para ello utilizaremos .la biblioteca SimpleXML de PHP; Requiere al menos la versión S.0.0 de PHP pero estll habilitada de forma predeterminada.

Utilizar SOAP con PHP

La otra interfaz que ofrece los mismos servicios Web es SOAP. Para acceder a ellos por medio de SOAP, tendremos que utilizar una de las diferentes bibliotecas SOAP de PHP. Existe una biblioteca SOAP incorporada pero como no siempre est6 disponible, puede utilizar la biblioteca NuSOAP, ya que al estar escrita en PHP no es necesario compilarla. Se trata de un solo archivo que se invoca a través de require_once (). NuSOAP esti disponible en http: / / dietrich. ganx•. com/nusoap, bajo Lesser GPL, es decir, la debe utilizar en aplicaciones que no sean gratuitas.

Almacenar en caché

Como hemos ——onado anteriormente, una de las condiciones que impone Amazon a los programadores es que los datos descargados de Amazon a través de

EL_HOLISTICO@HOTMAIL.COM - 2006

los servicios Web se deben almacenar en caché. En nuestra solución, tendremos que encontrar una forma de almacenar y reutilizar los datos que descarguemos hasta que se pase su fecha de utilización.

Presentación de la solución

En este proyecto volveremos a utilizar un enfoque dirigido a eventos para escribir nuestro código, como hicimos en capftulos anteriores. No dibujaremos un diagrama de flujo del sistema ya que &te cuenta con escasas pantallas y los enlaces entre las mismas son muy sencillos. Los usuarios comenzarán en la pantalla principal de Tahuayo (véase la figura 33.1).

Figura 33.1. La primera pantalla de Tahuayo muestra todas las caracterllflcas principales del sitio: navegación entre categortas. bílsquedas y el carro de ta compra.

Como puede comprobar, las caracterfsticas principales del sitio son la pantalla Selected Category y los artfculos de dichas categorfas. De forma predeterminada, en la página principal mostramos la categoría Current Best Sellers. Si un usuario pulsa sobre otra categoria, accederá a una pantalla similar para la misma.

Un pequef\o inciso terminológico antes de continuar. Amazon se refiere a las categorías como nodos de navegación. Verá que esta expresión se utiliza a lo largo del código y en la documentación oficial.

ta documentación proporáolla una lista parcial de los nodos de navegación mú populares. Ademú, si quieie uno en a,ncreto, puede acceder al ~itionormal de Amazon.com y leerlo desde el URL. pero no se puede obtener ~ lista completa.

Si se fija, verá que en la parte inferior de la p~ hay mú libros Y enla~ a páginas adicionales áe los que se pueden vt!ren esta una~. Mostraremos 10 libros por página, junto con enlaces a un múimo de 30 págmas. Este ':-!°r de ~0 por página lo establece Amazon. ~ ~ de ~ páginas es una decw6n arbitraria pe-~ aquí, los usuarios pueden pulsar sobre la información detallada de cada libro. En la figura 33.2 se puede ver esta pantalla.

figura 33.2. La péglna de detaHes muestra Infonnaci6n adicional sobre un detarfflinaclo libro, Incluyendo produc:los similares y criticas.

Aunque no cabe en una sola figura, mostramos la práctica totalidad de la información que nos envía Amazon a,n una solicitud a esta página. Hemos optado por omitir las partes dirigidas a otros productos que no sean libros asf como la lista de categorias en las que se incluye el libro.

Si el usuario pulsa sobre la imagen de la portada, podrá ver una versión de mayor tamai\o de la misma.

Habrá reparado en el cuadro de búsqueda situado en la parte supen?~ e pantalla. Esta función ejecuta una búsqueda de palabras clave en nuestro 11tio Y en el catálogo de Amazon a travn de su interfaz de servicios Web. En la figura 33.3 se muestra un ejemplo del resultado que devuelve una de estas búsquedas.

EL_HOLISTICO@HOTMAIL.COM - 2006

FIgur• 33.3. Resultados de la búsqueda aardman.

Aunque sólo mostramos algunas categorías, los clientes pueden acceder a cualquier libro por medio de la opción de búsqueda así como desplazarse hasta libros concretos. úda libro tiene un enlace Add to Car't (Añadiz:a, carro). Al pulsar sobre este enlace o sobre el enlace Details (Detalles) se accede a los contenidos del carro, como se muestra en la figura 33.4.

Por último, cuando un cliente pasa por caja al pulsar sobre uno de' los enlaces Checkout, enviamos los detalles de su carro de la compra a Amazon *y* le llevamos hasta allí. Accederá a una página similar a la de la figura 33.5.

Con esto debería comprender lo que queremos hacer al generar nuestra propia interfaz de usuario y utilizar Amazon como sistema.

Como vamos a utilizar de nuevo el enfoque dirigido por eventos, la mayor parte de la lógica de toma de decisiones se encuentra en un solo archivo, index. :php. En la tabla 33.1 se resumen los archivos de esta aplicadón.

T•bl• 33.1. Archivos de la apliéadón Tahuayo.

	Aplicación	El archivo de aplicación prir
	Aplicación	Muestra la página About
	Archivo de inclusión	Define constantes globales

topbar.php	Archivo de inclusión	Genera la b8rr8 da i1tonnecl6n que mparece MlaP911esuperiardellllel a• ——Yla CSS
bott-.pllp	Archivo de inclusión	Gener8el pledap6giriadetodal•~
AmazonRe•ultSet.php	Archivo de clases	Contiene la daada PHP que IImacen• el multado de c:acta una da la COMUIIH Amamn
Product.php	Archivo de clases	Contiene la c:1ae de PHP que .imac:-informad6n sobre undetenniNdOIIbftt
bOOkdisplayfuncr.ions. pbp	FundonN	ConlillNe funcionN qua permilen mostrar un Ubro y una lilla da Mbrol
cacbetunction•.php	Func:IDMs	ContianefunclonN .,..._ rullzarelalm_. namiento en CIIChtniquarielo por Amazon
cartfunctiODS.pbp	Func:ioMs	Contiene func:IDMs retacionaela con el carro de la compra
categoryfunctions.php	Funciones	Contiene func:IDMs que P!lffli•n recuperar y mostrar una catagorla
utilltyfunction•.php	FuncionN	Contiene una c:anjunto de funcioMS da ulill-dad que se udlizan alo largo de laapllcaCi6n

Figura 33.4. Deede la pégina del cano de la compra se pueden borrar artlculos, vaciar el carro o pasar por caja.

EL_HOLISTICO@HOTMAIL.COM - 2006

Figura 33.5. Loa artlculoa que estaban en el carro del cliente en Tahuayo aparecen ...,,. 811 W-ellmtde AIMl!eR.

Tambim necesitará el archivo nuaoap •php que mencionamoe anteriormente, ya que es obligatorio en estos archivos. NuSOAP se encuentra en el directorio del CD-ROM correspondiente a este capítulo, pero puede reemplazarlo por la nueva veni6nqueencontraráenhttp://dietrich.ganx4.com/nuaoap/ind.ex.php.

En primer lugar analizaremos el archivo de aplicación principal, index. php.

Aplicación principal

El archivo de aplicación index. php se muestra en el listado 33.3.

U... do 33.3. Index.php. Archivo de apllc:acl6n principal.

```
c?php
11 utilisaremoa GnicaMnta una variabla de •••1611, 'cart•, para alucanar
11 loa contanidoa del carro
••••ion_atartlll

require_once('conatenta.php'11
require_onceC'Produc:t.php'11
require_onceC'Allas0111.eaulcSat.php'11
require_once(•11tilityf1&DCtiona.php'11
require_once1'booltdiaplayf1&DCtiona.php'11
require:once('cartf1&Dctioáe.php'11
```



```
require_aac:e(•c:ate,oryfuaction_e.pllp'11

// ietae HD lU ,radablee que eeper- del uterior'
// se validarAn y convertirAn en global••
$uteraal • array('action', 'MDI', 'llllde', ·br-ellode', 'paga' '
'aearcll');

// lae Tariabl•• pueden venir de aet o Pollt
// convierta todaa lu variable• uternu eaperaü9 a II0\dlr•• global•• corto•
foreacll ($utemal ae $el
(
   if(8$llmQIISST($ell
      $$• ~ $ llmQIISST($ell
   •]••
      $$e• "1

   $$e • tria(S$el

// valoree predeterainadoe para lae variabl.. global••
ifl$-4e• •''I
   $llllde • •boob •, // llo ee han probado o~roe mdoe
if($broweellode• •• ')
   $br-ell0de • 531 // 53 correaponde a libro• eupenentae que DO - de ficcidn
if($pag• • ••' ,
   $paga. 11 JI Pri • ra pagina· laay 10 artlculoe por p&gina -;

// validar/recortar entrada
iflleregi('ªlA·IO·tJ+$',  $ASDII  // loe MI • debea eH aUen~iooe
   $MI•  •"1
if(leregi(•ª(a·•I•$',  $modell  // el modo deba eer alf.Wtico
   Smod• • •boolc•'
$p&ge•intval($page)  //  lae p&ginae y loa b~-ode  deben eer enteroe
$broweellode • intvall$browe ..odel1
// puede reeultar canfuao, pero eliaina1111a caracter•• da
// $aearchl •• aconaejable modificarlo ahora para que ae — • tre en •1 tltulo
$aearch • eafe1Jtring($aeatchl1

ifllieaetl$_SSSSIOll['c:art'111
(
   ••••ion_regieter('cart'11
   $_SBSSIOS('cart') • array(11


// tarea• que debe realizar ante• de aoetrar la barra •uperior
if($action • • •addtocart'I
   addToCart(S SBSSIOS('cart') , $MI•, $110da1
ifl$action • .-•deletefroacart'I
   deleterro.cart($ SSSSIOS( 'cart' J, $Ml•I
if($action • • ,•aaptycart'I
   $_SBS8IOII('c:art'I • array11

// aueatre la barra euperior
require_onca ('topbar.php'11

JI bucle da —— toa principal. 1eacciona a accione• del uuario en la
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
// p.tgiaa de 1avocac:1611
switc:b (Sac:tioal
(
   case 'detall' :
      ebowCategoriee(SIIOd•11
      sbowDetail(SAIIB, SIIOdel:
   break,

   case •acldtoc:art• :
   c:a•• •deletefra11c:art•
   case •emptyc:art• :
   case 'sbowc:art• :
      ec:bo 'chr />cbl>Your Sbopping Cartc/bl>':
      abowCart(S_SIESSIOB['c:art'l, SIIOdel:
   break;

   c:a•• 'image•
      ebow<:ategoriee($110del1
      ec:bo •cbl>Large Procluc:t I.agec/bl>':
      •b—I-ge(SASIB, $moda):
   break;

   c:aee 'eearc:b' :
      eboWCategoriee(SIIOde):
      ac:bo •cbl>learc:b Reeulte For •seearc:b•c/bl>":
      eboWSearc:b(Seearc:b, Spage, $.acle):
   break;

   c:aee 'br—••nocle':
   default•
      abowCategoriee(SIIOd•I:
      Sc:ategory • 9etCategorylla-(SbroweelfOcl•I
      ifOSc:ategorytlSc:ac.990"rY-•' ..•t Selliag IIOOlce')
      (
         ec:bo 'cbl>CUrreat Beet Sellersc/bl>':

      •|••
      (
         ac:bo •cbl>CUrreat leet Sellara in Sc:ategoryc/bl>":
      t
      abowarOW•ellode(Sbroweellode, $paga, SIIOdel :
   break;


require ('bottoa.pbp›,
? >
```

Analicemos este archivo. En primer lugar crearemos una sesión. Almacenaremos el carro de la compra del usuario como variable de sesión, como hemos hecho anteriormente.

Tns ello incluiremos varios archivos. La mayoría son funciones que describiremos mú adelante, pero n-oe detendrem.c» en el primem de ellos. Este archivo, coa•tanta#pbp, define importantes constantes que se utilizarút a lo largo de la apliad6n. Los contenidos de con•tants. pbp se muestran en el listado 33.4.

Llsmdo 33..4. constants.php. Declarac:lón de vñbles y c:anstantea globales eleve.

```
c?pbp
//••puede realiaar la c:oaex16D a aeta aplic:ac:i611 par aeclio de REST (XIIL
11 eobn BTTPI o 80AP
// defina la vera1611 de 11ftROD que quiera ••lec:c:1-r.
define('ldTIIOD', 'IOAP'):
defiu,•QfflOD•, 'RUT' 11

// aeegOn•• de c:rear•l directorio c:a~. para que .. pueda ••c:ribir
defiu('CACIIB', 'c:ac:be'); // ruta a loe arc:bivoe a c:ac:b•
defiu('USOCIA'l'BID', '-b ..nic:ee-20): //aluda .qiaf. IIII td. de aeoc:iado
defiu('DSITMI', ~IDIOIIIIIICI'); // dada -.¡uf. au etiqueta de deurrollador

//proporcione UD error el .. ejecuta el software CODla devtag fic:tic:ia
if ‹DIIVTIG• •'xxxxxxxxxuxxx '|
{
   die (•You need to eip up for aa Amaaoa.c:oa developer tag atea href •
      •bttp,//aeaoc:iatea ....aoe.c:om/exec:/paa .../aaeoc:latee/joia/
      developer/applic:atioa.btml/ref••c:_bb_l_0/002">AÍIA•onc/a>
      wbea you inatall tbie eoftware. You sbould probably eip up
      for u aaeoc:iate ID at tbe •- ti—. Bdit tbe file c:oaetuate.pbp. •) 1


// lista (parcial) de brownllode de Amaacm.
Sc:ategoryLi•t • array(S•>'Collputer• • Internet•, 3510•>'Web Devel~nt•,
         295223•>'PRP', l?•>'Literatun aDd Fic:tiOD',
         l•>'BueiAeee • In,,.etiDCJ', 53•>'110a Plc:tioa',
         23•>:ao.ance•, ?S•>'Scienœ•, 21.•>'Refereace.
         & •>Food • Wine•, 2?•>:Travel:,
         1C2?2•>'Sc:ieac:e Fic:tiOD'
         )!
```

Esta aplicación se ha desarrollado para utilizar REST o SOAP. Puede indkar cuál va a utilizar si cambia el valor de la constante MBTBOD.

La C011Stante CACHB contiene la ruta a la cacM de los datos que descargaremos de Amazon. Puede cambiarla por la ruta que quiera utilizar en su sistema. La constanteASSOCIATBD almacena el valor de su Id. de asociado. Si lo envía a Amazon con transacciones, recibirAuna comisión. Cmlbielopor su Id. de asociado. La constante DBVTAG almacena el valor del identificador de programador que Amazon le asigna al registrarse. TendrA que cambiarlopor su propia etiqueta de programador o la aplicación no funcionan. Puede solicitar una etiqueta en http://www.&111&son.com/gp/aw•/landing.btml.

Volvamos a index.pbp. Contiene algunos preliminara uf como el bucle de eventos principal. En primer lugar extraemos todas las variables entrantes de la matriz superglobal $_RBQOBST que provengan a trav& de GBT o POST. Tru ello, definimos valores predeterminados para ciertss variables globales ~dndar que determinan qui se mostrad poeterionnente,como se indica a continuación:

```
// valoree predeca:i;miaado• para lae variable• global••
ifls-dea •")
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
   $,IM)de • 'booke' I //  llo ee ban probedo otro• IIOdoe
if ($broweeWode• •''1
   $broweellode • 531 //53 correeponde a libro• •uparventH  que no eon de ficci6n
if($page• •' ')
   spage • 11  // Primera pagine·  hay-SO art1culoe por pagina
```

Hemos definido la variable modecon el valor booka. Amazon admite otros muchos modos (tipos de productos) pero en esta aplicación sólo nos centraremos en libros. No debe resultarle complicado modificar el código de este capitulo si quiere trabajar con otras categorias. El primer paso consistirá en restablecer $mode. Tendrá que revisar la documentación de Amazon para ver qué otros atributos se devuelven para productos que no sean libros *y* eliminar el lenguaje especifico de los libros de la interfaz de usuario.

La variable browaeNodese utiliza para especificar qué categoría de libros queremos mostrar. Debe configurarse si el usuario ha pulsado uno de los enlaces Selected Categorles. Si no se configura, por e;emplo, cuando el usuario accede por primera vez al sitio, la definiremos como s 3. Los nodos de navegación de Amazon son simples enteros que identifican a una categorfa. El valor s 3 representa la categoría Non-Fictlon Books, que es la que mostramos en la página inicial.

La variable page se utiliza para indicar a Amazon qué subconjunto de los resultados queremos mostrar dentro de una determinada categorfa. La página 1 contiene los resultados del 1 al 10, la página 2 los resultados del 11 al 20 y asf sucesivamente. Amazon define el número de articulos por página, algo que no podemos controlar personalmente. Evideuteniente podríamos IIIOStrer dos o más "páginas" Amazon de datos en una de cada cuatro páginas pero 10 es un número razonable.

A continuación, limpiamos los datos que hayamos recibido, yá sea a través del cuadro de búsqueda o por medio de parámetros GBT o POST:

```
//validar/recortar  entrada•
ifl:eregi('4[A-Z0-9)+$',     $ABI•>I  // lo• ASI• deben •ar alfanum6rico•
   $ASIR •'
ift1eregi(••1a-sl+$,    $11C1dell  // el IDOdo debe Hr alfabttico
   $mode •  book•'1
$page•intval($pagal;   // la• pagina• y loe brow•ellode• deban •er anteroe
$brow•e•ode • intval($brow•e•odal;
// puede re•ultar confueo, paro eliainamo•  caracteree de $eearch,
// e• acon•ejable modificarlo ahora para que •e mue•tre an el titulo
S•earch • •afeString!$•earchl  ;
```

Nada nuevo. La función aafeString C> se encuentra en el archivo utilityfunctiona. php. Simplemente elimina todos los caracteres que no sean alfanuméricos de la cadena a través de un reemplazo de función convencional. Como ya hemos mencionado esta operación anteriormente, no la describiremos.

La razón principal por la que tenemos que validar las entradas es que utilizaremos las entradas del usuario para crear nombres de archivo en la caché. Podrfamos tener serios problemas si permitiéramos que los usuarios incluyeran .. o / en sus entradas.

Seguidamenteconfiguramos el carro de la compra del usuario, en caso de que no tenga uno:


```
iflli••et($ _SBSSIC11['cart')II
(
   •e•elon_regi•ter('cart');
   $_SISSIO11('cart'J   • array(I;
```

Antes de poder mostrar la información en la barra superior de la página (véase la figura 33.1) tendremos que ocuparnos de un par de aspectos. En esta barra superior se muestra una imagen del carro de la compra. Por lo tanto; conviene que la variable cart esti actualizada antes de poder representarla.

```
// tarea• que deban realisar•e ante• de aoetrar la berra •uparior
ifl$action • • 'addtocart•)
   addToCart($ SISSIOII('cart'),  $ASIR,  $11C1del;
if($action • .-•cteletefro11Cart•)
   deleteFromCart($_SISSIO11['cart'J,   $ASIN11
if($action • • 'emptycart•)
   $_SBSS10N('cart'J  • array11;
```

En este caso, según sea necesario, af\adimos o eliminamos articulos del carro antes de mostrarlo. Volveremos a estas funciones cuando analicemos el carro de la compra *y* el pago. Si quiere verlas ahora, las encontrar, ~ el uchivo cartfunct iona . php. Por el momento las dejaremos de lado ya q.. primero conviene entender la interfaz de Amazon.

Tras ello, incluimos el archivo topbar. php, que simplemente contiene HI'ML, una ho¡a de estilo *y* una sola invocación a la llamada ShowSmállCart O (de cart funct iona . php). De esta forma se muestra el pequeflo reswnli!I'del carro de la compra que aparece en la esquina superior derecha de las imágenes. Volveremos a analizarlo cuando describamos las funciones del carro.

Por último, llegamos al bucle de control de eventos principal. En la tabla 33.2 se recoge un resumen de las posibles acciones.

Tabla 33.2. Posibles acciones en el bucle de eventos prlnclpal.

browsenode	Muestra libros de la categorla especificada. Es la acción predeterminada.
detail	Muestra los detalles de un determinado libro.
i-g•	Muestra una versión ampliada de la portada del libro.
aearch	Muestra los resultados de la búsqueda de un usuario.
addtocart	Allade un articulo al carro de la compra del usuario.
deletefroacart	Elimina un articulo del carro de la compra del usuario.
eaptycart	Vacla el carro de la compra•
ahowcart	Muestra los contenidos del carro.

EL_HOLISTICO@HOTMAIL.COM - 2006

Como puede comprobar, las primeras cuatro acdones de la tabla estm relacionadas con la recuperación y la representación de información de Amazon. Las cuatro siguientes est6n relacionadas con el carro de la ~ompra.

Todas las acciones que recuperan datos de Amazon funcionan de forma similar. Veremos un ejemplo en el que se recuperan datos de libros en una determinada categoría (browaenode).

Mostrar los libros de una categoría

El código que se ejecuta cuando la acción es browsenode>(ver una categoría) es el siguiente:

```
llbowCategorieIlt$110de);
$category • getcategoryllame($browaeNodel;
if(t$categoryllScategory•  •'leet Selliag Boolca'I
{
   ecbo 'cb1>CUrrent leat Sellerac/b1>';
}
elae
{
   ecbo •cbl>Current •••t Sellara in Scategoryc/bl>";
}
a~BrowaellodelSbrowaeNode, $page, Smodel;
```

La función ahowcategories (I muestra la lista de categorfu selecdonadas que apar~ en la parte superior de la mayoría de las p6ginaa. La función getCateg9..ryName () devuelve el nombre de la categoría actual si se proporciona su nmn~ browaenode {). ~función ahowBrowaeNode () JDUestra una p~gina de libros en dicha categoría.

En primer lugar nos detendremos brevemente en la función ahoWCategoriea (I. El código de esta función se recoge en el listado 33.5.

Uatado 33.5. Función showCalegories() da catago,yfunclions.php. Una lista da calegorlas.

```
//mueatre una liata inicial de categorlaa popular••
functioa abowcategoriealSIIOdel
{
   globel $categoryLiat;
   ecbo 'cbr />cb2>Selectecl categorieac/b2>1

   ifCSIIOde • • 'booka•)
   I

      •aortfScategoryLiat);

      tcategoriea • count($categoryLiat)1
      tcol-•     • 4;
      Srow•.• ceill$categoriea/$colW11Dal;

      ecllo 'ctable bordar •"O" cellpaddiag •"0" cellapaciag·•o• widtlt
- ·loot•>ctr>1
```



```
reaet|ScategoryLiatl;

for(Scol • O; $colc$colllllll1al $col++)
{
   IICbo 'ctd widtb • "'.(100/$colUllllll).•t•  vali9D • •top°>cUl>'t
   for(Srow • o, $rowc$rowe, Srow++I
   {
      $category • eacb(Sc:ategoryLiatl;
      if 1$category)
      (
         $brow..lloda • Scategory('key'J;
         •—   • Scategory('value'J;
         echo •cli>c9P&a claaa • 'category'>ca bref •
               'iadex.pbp?actioa•broweaode.t»row11ellode-
               Sbrowaelloda'>$nal111c/a>c/apaa>c/li>"1


   }
   echo 'c/ul>c/td>';
)
ecbo 'c/tr>c/table>cbr />';
```

Esta función utiliza una matriz denominada categoryLi at (l declarada en el paquete categoryfunctiona. php para asignar números browaenode a nombres. Esta función ordena la matriz y muestra las diferentes categorfas.

La función getCategoryName () que se invoca a continuación en el bucle de eventos principal se utiliza para buscar el nombre del browsenode en el que nos encontramos, para representar un tftulo en la pantalla como por ejemplo Current Best Sellers in Buainea. &·Investlng. Como-hemos mencionado anteriormente, lo busca en categoryList. La diversión empieza al llegar a la función showBrowaeNode (l, que mostramos en el listado 33.6.

Llat.do 33.1. Función showBrow1eNoda() de bookdlsplayfunclions.php. Una llst. de categorlas.

```
// Para un deterainado browaeaode, muellre una pAgina de producto•
fuactioa abowllrowaeNodel$browaellode, $p119e. SIIOdel
(
   Sara • tjetJlllS (•br—   ', array('brow9eaode'•>Sbrowaellode,
                                 •paga••>  Spage,
                                 'IIOde' •>$110de))1
   abowl —— ry(Sara->productal), $paga, Sara->totalaeaultaO,
         Seode, $bronellodal;
```

Esta función realiza dos operaciones. En primer lugar, invoca la función getARS (l de cachefunctions. php. Esta función obtiene y devuelYe un objeto AmazonResultSet (como veremos mú adelante). Tru ello, inv~ la función ahowSummary() de bookdisplayfunctions. php para mostrar la infonnaci6n recuperada.

EL_HOLISTICO@HOTMAIL.COM - 2006

La función getARS(› ee fundamental para controlar la aplicad6n. Si analizamos el resto de las acciones del código como las de ver detalles, im4genes y búsquedas, comprobad que todu vuelven a la misma.

Obtener un clase AmazonResultSet

A continuación analizaremos la función getARS() con °"8 detalle. Su c:6digo se recoge en el listado 33.7.

Uatado 33.7. Funcl6n getARS()ele cachefuncllona.php. Con¡unto de ruultados de una COIIIU ....

```
// Ollteaga un AaasollllleaultSet da la cae.. o de wua coaaulta
// Si ae trata de una conaulta, aUdala a'la cachf
fuaction getAIIS($type, $para..tara1
(
   $cache• cached($type, $par...teral1
   if($cache) // ai ae encuentra ea la ca~
   (
      retura $cache,
   }
   elae
   (
      $alC& • IleY AmasOlllleaultSet1
      if($type • • 'aaia')
         $ara·>ASIKSearch(padASIM($par...tera(•aaia•1), $para..tera['moda'll1
      if($type • • •browee•)
         $ara•>browaellode8earch($par..tera['broweenode'I, $par...tasa[•page•J,
                                $para..tara('IIClde'lll
      if($type • • •aearch')
         $ara·>keywordSearch1$para..tara(•aaarch'1, $par...tera[•pave•J,
                               $para..tera['IIClde')I;
      cache($type, $par...tera, $aral1

   retura $ara;
```

Est6 función se ha clisdado para controlar el proceso de obtención de datos de Amazon. Se puede realizar de dos formas: bien desde la caclw o directamentedesde Amazon. Como Amazon requiere que los programadores almacenen en caché los datos descargados, la función busca primero los datos en c:adw, como veremos más adelante. Si todavía no hemos ejecutado dicha consulta, sen necesario obtener los datos directamente de Amazon. Para ello creamos una instancia de la clase AlllasonReaul tSet e invocamos el método correspondientea la consulta concreta que queramos ejecutar. El tipo de consulta lo determina el panlmelro $type. En el ejemplo de búqueda de categoría, pasamos browae como valor de este parámetro (véase el listado 33.6). Si queremos realizar una consulta sobre un cleterm:inado libro, habri que puar el valor asin. y si queremos realizar una búsqueda de pala· bru clave, estableceremos el parúnetro como search.

Cada uno 'de estos par,metros invoca un método diferente en la clase AllasonResultSet. La búsqueda de un artfculc;, concreto invoca el método ASIHSearch(), la bdsqueda de categoría llama al mélodobronellodesearch() y la búsqueda de palabru clave, al método lteywordSeach(). En el listado 33.8 se incluye el c:6digocompleto de esta clase.

Uatado 33.1. AmazonReaultSet.php. Clw para proceur couexlal• a Amazon.

```
c'l'php
// pueda ca.titar entre RBST y SOA1' por medio de aata conataata definida ea
// cona.:.nta.php
if(IISTROD• •'SOAJI')
.(
   iacluda_once(•aúaoap/auaoap.php•)1


// Bata claaa almacena el reaultado da conaultaa
// Suele aer 1 o 10 iaataaciaa de la claae PrOduct
el••• AllasonReaultSet
(
   private $brovaell0de;
   private .$page;
   private $11Cldel
   private $url1
   private $type;
   private $total Reaulta;
   pdvate $Curreat.2rodllct • DUll;
   private $producta • array(l1 // matria da objetoe Product

   fuaction productall
   (
      ratura $thia·>producta1


   fuactioa totalReaulta()
   (
      retura $thia->totalReau~t•1


   fuactioa getProduct1$11
   (
      if(iaaet($thia->producta($illl
         retura $thia->producta[$il
      •la•
         retura falae;


   // Realica wua conaulta para obtener una pigiaa completa de producto•
   // de un browaeaoda
   // Cambie entra XML/ll'l'TP y SOAP aa conatanta.php
   // Devuelve una .. tris de Producta
   fuaction brOV8dodeSearch($browaelfode, $page, $110del
   (
      if(ll&THOD• •'SOAJI')
```

```
SaoapclieDt • Dev aoapclieDt!
   ~ttp://aoap .... zoD.C09/ac:!le...2/AmazODN~ervicea.vadl•,
   'vadl')1
S•oap_proxy • $aoapclieDt·:,,getProxyl)r
Spar    tera('IIOde'l•$model
Spar    tera(•page•J•SP-,e1
Spar    tera('type•]•'beavy'l
Spar    tera('ta9')•$thi•·>•••ocID1
Spar    tera('devtag')•$thi•·>devTag,
Spar    t•r•('aort')•'•••learaak•l
Spar    t•r•('brov••-nod••J •  $brovaellode,

// ejecute la conaulta •oap
Sreault • $aoap_proxy->Brov•ellodeSearchllequeat1Spar...t•r•>,
ifli•SOAPlrror(Sreault!)
   return falH ı
Sthia·>totalaeeult• • $reault('Totala.•ulta•J,

for-ch(Sreault C'Detail•' J •• $product)
(
   Sthia·>producta() • nev Product ($productl;
}
un••~(Saoapclient);
unaet<S•oap_proxyl,
}
•]••
(
   // czee el OU. • iDWOqUe paraellllL para de•cargarlo y aaalizarlo
   leM• seype ‹bzoffaa ,
   Sthi•·>brovHllode • Sbrov•ellode;
   Sthia·>page • $paqa,
   Stlia·>.ade ;; smoclel .....
   Sthia·>url • 'http://Xlll .... zon.c09/onca/xal.Z?t•'.ASSOCIATIID
              •'Mlev·t• ••DaVTAG. •Urov•elfOdesearch-'
              .$thi•·>brovaesode.'&IIOde•'Sthia·>  IIOde
              ·'•type•heavy&page•'-Sthi•·..pmge.'&eozt•••alearaak•f•xml.'1
   Sthi•·>par•ellNL()1

return Sthia·>producta,

// C'oll UD AIIS dado, obtenga el UU. de la iM9- de gran ta..ao
// Davaeln una cadena
function getI-.e11rlLarge(SABIS, $110de)
(
   foreach($thi8->product• aa Sproduct)
   (
      if( Sproduct·>ASIS()• • $A.SIS)
         Htllffl Sproduct·>i-,ellRLLarge(l ı
   }
   // •i DO eJd.ate
   SU&ill·>AIDISe&rcla($UIS, S-..J I
      return Sthia->prodacta(Ol~>i~(I,
```



```
// a.alice una consulta para obtener UD producto con el AIIS ••pacificado
// callbi• entre XNL/IITTP y sea, an coaetanta.pllp
// D-lve UD objet:o Prodact•
functicn ASISSÑrch(SASIS, $moda • 'bcloka')
(
   Sthia·>type • •Mm;
   $thi•·>ASIS•$ASIS!
   $thia·>90de • Smade!
   $A.SIS • padASIS($ASIS) ı

   if (IIftIIOD• •'SOAP' I
   (
      error_reporting(I_ALL • -B.IIOTICI);
      $aoapcliant • _nev aoapclient (
         'http://aoap .... zaa.:c:oia/•ch..-•Z/Amazaa.llabServicea.vadl',
         •vadl') ;
      $—p_proxy • Saoapcliant•.getProxy O;
      $par tera ('aain' I •$ASIS ı
      Spiar tera(•IIIDda•l •SIIOde,
      $para—ter•e. type•] •beavy•'
      Spara—ter•('tag')•Sthia·>_aaaocID;
      Spar...ter•('claYtag'l•Sthi•·>_de.,,._.,

      // •:tecute la conaulta aoap

      $reault • Saoap_proxy·>binSearchaeque•tlSpar-tera);
      if(iaSOAPlrror(Sreault))
      (
         print_r(Sreeult);
         zetma falsa,
      }
      Sthi•·>producta(0I • nev Product1(Sr11ault('Deta~l•'l(Ol)l
      Sthi•·>totalRewlt••lr
      unaet(Saoapcliantl;
      Wia~t(Saoap_proxyl;

   •]••
   (
      // crH el URL • invoque par•eXNL para deec-.rgarlo y analizarlo
      $thia·>url • 'http://xal.amazon.ca11/onca/Xlll.Z?t•'.AISOCIATIID
                 .•Mlev·t•' .DBVTAG. '~1DS-rch•'
                 .$th1•·>_ASIS
                 .••type•heavy•f•Xlll';
      $thia->par•eXIIL(l1
   }
   return $thi•·>11roducta(Oll

// B:tecute una CODaulta para obta8r una ~ina C011Pleta de prodactoa
// caa. una bllaquecla cle palabra• clave
// cabie -tre JINL/IITTI' y sa.p en 1 ~ .php
// DeYuelve una •triZ U Prodm:t•
funceicn keyvorUearchlS•earch, $,..,., S-- ~ 'boalls'I
(
   1f (IISTBDI)o, • ! SOU")
   (
```

```
error_reporting(&_ALL • -a_l10TIC:SI;
$•oapclient  • nev •oapclient(
   'http,//•oap ....son.c:oa/•cbema•2/llllasonWebService•.v•dl','~••dl•)
$•oap_proay  • $•oapclient-:,,getProxy(I;
$para..ter•('II0de']•$aode1
$par — tu• ('page•] ·S～ge1  ..... -.
$para..ter•('type']•"heavy•j
$parameter•(•t～'1•$thi•->a••ocID1
$para..ter•(devtag'1•$thi•->devTag1
$par    ter•1'•ort']•'+•ale•rank'1
$par    ter•('keyvord,)    $•earch1

// ejecute la con•ulta •oap
Sr••ult  • S•oap_proay->lteyvordSearchReque•t($para..ter•I,

if(i•SOAP&rror($re•ultl  I
   return fal••,

foreach($re•ult(•Detail•'1    a•  $product)
(
   $thi•->product•l1  • nev Product($product)1

$thi•->total Re•ult•  • $re•ult[•TotalR••ult•'1
un•et($•oapclient11
UDHt($•oap_proxy11
}
el••
(
   $thi•->type  • '•earch';
   ～⊖->eeazch•S•eaLch;
   $thi•->page  • $page1
   $•earch  • urlencode($•earch);
   $thi•->110de  • $111ode;
   $th18->url  •  'http://xml .....son.com/onca/xal2?t•'.ASSOC:IATBID
                  . • ～v-t•,,D&VTAG,•'1teyvordSearch•'
                  ,$eearch.'r..ode•'.$thi•->   II0de
                  , '•type•heavy•page• '      -
                  .$thh->_page
   '              ' '••ort•+•ale•rank•f •xml , 1
   $thie->parHXML0  1                          •

return $thi•->product•1


// Analice el XML en el objeto/objeto•  Product
function par•eXNL()
{

   // elimine lo• errore• porque •u•l•  fallar en oca•ione•
   $xal  • ••1mplexml_load_file($thie->urll1
   if(!$xml)
   (
      //pru6belo de nuevo en ca•o de que el ••rvidor ••tuviera ocupado
      $xml  • ••implexml_load_file($thi•->urll'
      1C(!$xml1
      (
         return fal•e,  '
```



```
}
$tbi•->total Re8Ult•  •  (integer1$Xlll->Total Re8Ult•;
foreacb($Xlll->Detail8 u  $pzadlac:UIIL)
(
   $tbi•->product•l1  • nev Product($prodactXNLl1
}
```

Esta clase tan útil encapsula la interfaz a Amazon en un cuadro de color negro. Dentro de esta clase se puede realizar la conexión a Amazon por medio de RESTo con ayuda del método SOAP. El método que se utilice viene determinado por la constante global MBTHODque definimos al principio.

Comenzaremos desde el ejemplo de bdsqueda de categorías. Utilizamos la clase AmazonResultSet como se indica a continuación:

```
$ar••  nev llllasonRe8UltSet1
$are->broweeJlodeSearcb($par ...tere(,brow.enocle,),
         $parameter•('page•J,
         $parameter•('II0cle,)l1
```

La c～ carece de constructor por lo que pasaremos directamente al método browaeHodeSearch(). Le pasamos tres parúnetros: el número browsenode que nos interesa (que se corresponde pm ejemplo a Buainus & IDYUllng o Computer8 I, Internet), el número de página (que representa los registros que qu~os recuperar) y el modo (que representa el tipo de productos que nos interesa). El código de este método se muestra parcialmente en el listado 33.9.

Llstlldo 33.t. M6todo browleNodeSurch(). B6squeda de categortaa.

```
function broweellocleSearcb($brov•..ode, $page, $110cle1
(
   if (MBTBOD• •'SOAP' I
   (
      $•oapclient  • new •oapclient(
         ,bttp://•oap.a  ..son.c:oa/•che ...2/A11asonWebService•.W8dl•,
         •v•dl');
      $•oap_proay • $•oapclient-:,,getProxy(I;
      $par    ter•('II0de,)•$110cle1
      $par    tere (•page• J •$page1
      $par    t•r•l'type'l•'beavy';
      $par — t er•l'tag']•$tbi•->a•eocID1
      $par    tere(,devtag']•$thi•->devTag1
      $par    tere(••ort')•'+•ale•rank';
      $par    tér•l'brov•e_node']  • $brov•ellocle1

      // ejecute la con•ulta •oap
      $re•ult  • $•oap_proxy->8row.ellocleSearcbReque•t($paraaeter•I  I
      if(i•SOAP&rror($re•ult11
         return fal••1
```

```
    $tbi•·>_totalaa•ult•  • Sraault['Totalaaault•'1l

    foraachl$ra•ult['Dataila•J  •• $productl
    (
       $thia·>_producta{)  • llaw Productl$productll
    )
    uaaat 1$aoapclia11tl,
    unaat($aoap_proxy>,
  )
  el••
  {
    // crea al uaL • invoque parnllNL para da•car9arlo v analizarlo
    $tbi•·>typa  • 'brov•a';
    $thi8·>brow••II0da • SbrowaaNoda!
    $thi•·>pag•  • Spaga,
    $thi•·>II0da  • s.oda,
    $thi•·>url  • 'http,//xml.amazo11.com/onca/xml2?t•'.ASSOCIATBID
              '',dav·t•'.DBVTAG.',arow•all0dasaarch•'
              .Sthi•·>  browaaModa.•,II0da•'.$thi•·>  moda
              '',typa•h••vv•P•V••'.$thi•·>_paga.',.ort•+aala•r•Dk•f•xml';
    $thia·>par•aXNLl11
```

En función del valor de la constante METHOD, este m~o realizará la consulta a travá de REST o a travá de SOAP. Veremos cada una de las opciones por separado.

Utilizar 'REST/XML sobre HTTP

En primer lugar definimos una serie de variables miembro:

- type: El tipo de búsqueda que se realiza. Buscamos libros dentro de un determinado broweenode, por lo que definimos el valor como browse.
- browse: El valor del browsenode concreto que hemos pasado como parámetro.
- ~age: El número de p,ginas que hemos pasado como parámetro,.
- mode: El tipo de artfculos que buscamos (por ejemplo books) que hemos pasado como parámetro.
- url: El URL en Amazon al que debemos conectamos para realizar este tipo de búsqueda.

Los URL para los que realizamos nuestras conexiones HTI'P para distintos tipos de búsqueda y parámetros se encuentran en elAPI de servidosWeb de Amazon.com y el paquete del programador. Veamos con mú detalle los parámetros GBT que pasamos en este caso:


```
Sthi•·>url  • 'http,//xml.amason.coa/onca/Xlll2?t•'·.1880CIATEID
            .'idav·t• ''DBVTAG.'urowaaNodaSaarcba'
            .$th1a·>_browaaModa.•,mod••'.Sth1a•>_.ada
            .',typa•baavy¡paga•''$th1•·>_paga
            .',aort•+•ala•rallklof•xml';
```

Los parámetros que tenemos que pasar a este URLson los siguientes:

- t: Su Id. de asociado.
- dev· t: Su identificador de programador.
- Browee!lodeSearch: El número de broweenode que quiere buscar.
- mode: Book• u otro tipo de producto v6lido.
- type: Heavy o l,i te. El primero devuelve mú información.
- page: Grupo de diez resultados .
- iiort: El orden enque queremos que se devuelvan los resultados. Es un parámetro opcional. En este caso, lo hemos configurado como +salesr.ank porque queremos los resultados en orden de ventas.
- f: El formato. Siempre contiene el valor 'Xlll '.

Entre los tipos vffidos de orden se encuentranJos siguientes:

- Artfculos destacados: +¡111rank
- Superventas:+salesrank
- Opinión media del cliente: +reviewrank
- Precio (de menos a m4s): +pricerank
- Precio (de más a menos): +inverse-pricerank
- Fecha de publicación: +daterank
- Alfabético (A-Z): +titlerank
- Alfabético (Z-A): -titlerank

Una vez configundos todos estos pamnetros, invocamos

para realizar la operación. El método pareeXNL(› se describe en el listado 33.10.

UatNo 33.10. M61odo parseXML(). AMIl.ia dll XML dewllfo por una consulta.

```
// Analice el XNL en al objeto/objeto.  Pz:oclucC
functiOD par••XNLCI
{
   // eli•ina lo• error•• porque •vale fallar en oca•ioD••
      $.mal.  ••1mpla.mal_load_fila($tbi•·>Urll;
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
if (1$-1)
(
   //pru&lelo de nuevo en ca•o de que el ••rvidor ••t-iera  ocupado
   $zal • ••ie,plezal_load_file($tbi•·>url)1
   ifl1$zal)
   (
      return fal••1


$tbi•·>totalR••ulte • (integer)$zal·>TotalReeult•1
foreacb($zal·>Detaile •• $productXMLI
(
   $tbi••>producteII • n•• Product($productXML)1
```

La función aimpleXML_load_file () se encarga de la mayor parte del trabajo. Lee el contenido XML de un archivo o, en este caso, de un URL. Proporciona a los datos una interfaz orientada a objetos y la estructura del documento XML. Es un interfaz lltil pero como queremos un conjunto de funciones de interfaz para trabajar con los datos recibidos a trav6s de REST o SOAP, podemos diseftar nuestra propia interfaz orientada a objetos para los mismos datos en instancias de la clase Product. En la versión RESTpodemos convertir los atributos de XML en tipos de variables de PHP. No utilizamos el operador cast en PHP, pero en este caso si no lo hicié~amos recibiríamos rep~taciones ~e obj_eto de_too.~ los datos, algo no demasiado útil. La clase Product contiene básicamente funciones de acceso para acceder a los datos almacenados en sus miembros privados, por lo que no es recomendable imprimir la totalidad del archivo. No obstante si nos detendremos en la estruetura de la clase y del constructor. En el listado 33.11 se recoge parte de la definición de Product.

Ualado 33.11. La clase Product Incluye Infonnacl6nsobre un producto de Amazon.

```
el••• Product
(
   privete $ASIR1
   private $productRe..,
   private $rele..eDete1
   private $-.nufacturer1
   private $i..geUrlMediW1l
   pr1vate $i..geUrlLarge1
   Fr1vate SliatPrice1
   pr1vata SourPrice,
   priv•c• :aalealla.Dk;
   pr1vate =•vallabiiity,
   pr1vate :av9Cuatoia.rRating,
   privat~ Sauthora.  array()i
   private $rev1ava.  •rrayll,
   private Sai=1;arPro4uct•. array(li

function __con•tr-Jct1s,...1l
```



```
if (IIBTIIOD••'IOIP')
(
   $tbi•·>A8Ilf • $Jml['A8iD)1
   $tbi•·>prodllctll.-. • $zal[ıProductRa..•J1
   U 1$zal[•Allltbon •1>
   (
      foreacb($Jml['Autbor•'I H  S•utborl
      (
         $tbi•·>autllor•ll • S•utborl
      )
   )
   $tbi•·>rel....nat• • $..l[ıRel•••eDate•J1
   $tbia->IIUIUfactazer• $xml[ıNanufacturer•l1
   $tbia·>imegeUrllledi.. • $..l(' I..geUrlNedi..'J ı
   $tbia·>il1&911U'rlLarge• S.. 11•1..geUrlLargeıJ1

   $tbia·>liatPrice • ...l['Liattrice'l1
   $tbia·>liatPrice. atr_replace('$', ••, $tbia·>liatPrice)1
   $tbia->liatPrice. atr_replace(•,', "  $tbia·>liatPricel1
   $tbia·>liatPrice • floatvall$tbia·>liatPrice)1

   $tbia->OUrPrice • $--1['Ollrtric•'l1
   $tbia·»ourtrice • atr_replace1•$', '', $tbia·>OUrtricet~
   $tbia-»ourtrice. atr_replace(•,•, ••, $tbia->OUrtricel~
   $tbi•·»ourtrice • floatvall$thia·>ourtricel1

   $tbia·>•aleallmlk • $xml[•Sale•Ranlt')1
   $1:1li1·>&YaUmility  • $-1(ı~lebilkyıJ1
   $tbia·>•Y9CU•tomerRating• $zal('Revieva•]['AvgOlatomerRllting•J1
   Sre•i..Count • o,
   if($zal[•aa.1...•][•cuatoaerRevie••'l)
   (
      fonacb  ,e..u•a.viwa•]  c•cuatC1918rRaYi•••'l •• erev1..,
      (
         $tbia->revi...[$revi..COunt)[•rating') • $reviw(•&atingJ ı
         $thi•·>nvieva[$nvievCount)l'•-ry')  • $revi..['S-ry'I  I
         $thi•·>rffi...[$revievCount)['cC11111ent•] • $reviev(•CC11111ent'l1
         $revi..ccwat++1
      )
   )
   ifl$zal('Biailartroduct•'l)
   |
      foraacb ($xal['SillilarProduct•'l •• $aiailar>
      (
         $tbia·>•illilartroduct•ll • $aiailarl


alee// ut~lice IIBST
(
   $tbia·>ABIII • (atring)$:11111->Asin1
   $tbia·>productR...  (atring)$xal·>ProductR...,
   if($zal·>Autllora·>Autbor)
   (
      foreacb($xal·>Autbora->AutborH  $autllorl
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
        $thi•·>&Uthorall  •  (atring)$autbor,

    )
    $thia•>releaaeDate •  (atriD9)$xml·>llele...Date;
    $tbia•>11&11llfactunr •  (atril19)$xml·>Nllnufacturer;
    $tbi•·>imageUrlNediwa •  (atriD9l$xml·>Ima99UrlNediua;
    $tbia•>imageUrlLarge  •  latriD9)$xml·>I11ageUrlLarge;

    $tbia·>liatPrice  •  (atring)$xlll·>LiatPrice1
    $thia·>liatPrice  •  •tr replace('S',   '',  $thia·>liatPricel1
    $thi•·>liatPrice  •  •tr:replace(•,•,   '',  $tbia·>liatPricel1
    $thia·>liatPrice  •  floatvall$thia·>liatPricel,

    $thia·>0urPrice  •  (atringl$.xal·>OUrPrice,
    $thia•>ourPrice  •  atr replace('$',   ••,  $thia·>OUrPricel,
    Stbia·>ourPrice  •  •tr:replacel',',    '',  $thia·>ourPricel;
    $tbia•>ourPrice  •  floatval($thia·>ourPricel;

    $thia·>••l..Rank  •  (atringl$xal·>SaleaRalllt1
    $thia•>&Vailaltility •  (atril19) Sxml•>IWailaltility;
    $thia·>avgeu.toiaerRating •  lfloat)$xml•>lleviev•·>AvgCuat011erRatiDg1
    $reviltllCOUDt.  o;
    U1$xml·>llevieva->C:U•t — rReviev1
    {
       foreacb  ($xml·>Reviewa->CllatomerReviev ••  Sreviev)
       {
          Sthia·>reviev•l$revievCountl['rating•J  •  (float)$review->Rating;
          $thia·>reviev•l$revievCountJ['au.mary•J  •  (atring)Sreviev-
       >SU11m&ry1
          $thi•·>reviev•l$revievColmtl [•c — nt'I  •  (atring)Sreviev-
       >C-nt;
         '$revievCOU11t++;

    )
    if($xml·>SillilarProducta·>Product)
    {
       foreach !Sxml·>SiailarProducta->Product ••  Saiailarl
       {
          $thie•>eiailarProdacta(J  •  latring)Seiailar;
```

Este constructor puede adoptar dos tipos de datos de entrada y aea una interfaz de aplicad6n. Aunque parte del código de procesamiento podría ser Dl'5 genérico, algunos atributoe tienen nombres difer~tes en función del método.

Una vez realizado todo este procesamiento para recuperar loe datos, devolvemos el control a ~ función getARS() y de ahí a ahowBrOW11eNo¢le. El siguiente puoes

```
llhov9....  ry(Sar•·>product•(I, Spage,
S•r••>totallleault•II, $aoda,
Sbr-•alloda1;
```


La fund6n ahowS - - —ry () simplemente muestra los datos de AlnazonResultSet, como puede comprobar en la figura 33.1. Por esta razón no hemos incluido aquí dicha función.

Utilizar SOAP

Volvamosa la versión SOAP de browaellodeSearch (). A continuad6n repetimos esta parte del código:

```
S•oapcliant • nav aoapcliant(
      'http://aoap.aaaaon.coa/achemaa2/AllaaonWabSar,,icaa.wadl','vlldl'I;
$aoap_proay • $aoapcliant·>9etProxy()1
$par...tere('.ode'l•$410de:
$para..tara(•page'l•Spaga;
$par...tare(•typa'l•'heavy•,
$paraaatare('tag•J•$thi•·>_aaaocID1
$par   tare('devtag'l•$thia·>_deYTag;
$par   tan(•aort'I•'•••l••rank•;
$par   tan(•b-••-DOde'I • $brovaalloda1

//ejecute la con•ulta aoap
$r—lt • Saoap_proxy->Br — ellodeSearc:!1119quut($par ...tar•I;
if(iallCIAPSrror($raault1l
   ratun  falaa;
$thia·>totallle•ulta • $ra•ult('Tota1Raaulta•J;

foraacl11Sn•ult('Datail•'I   •• Sproductl
{
   Stbi•·>ProductaCJ  • uw  Product($product):
|
uneat($aoapcliant1;
UDaatl$aoap_proxyl :
```

No hay funciones adicionales que analizar; el cliente SOAP se encarga de todo por nosotn,s. En primer lugar creamos una instancia del cliente SOAP:

```
$ — pcliant • aaw aoapcliant(
                 'bttp:// — p.aaaaon.coa/acb  ....2/Allaaonlfabller,,ice•.•adl',
                 '11adl);
```

Pmporcionunos al cliente dos panmetros. El primero es la descripdón WSDL del serricio y el segundo padmetro indica al cliente SOAP que se trata de un URL WSDL También bastaría con proporcionar un pamnetro: el punto final del servido, que es el URL directo del servidor SOAP.

Hemat optado por hacerlo de esta forma por una buena razón, como se indica en la siguiente linea de código:

```
Saoap_p:rosy • Saoapcliant·>9atProxyl11
```

Esta línea aea una clase en función de la información del documento WSDL Esta clase, el proxy WSDL.dispondr6 de mitodos que se co.aesponden a loe m6to-

EL_HOLISTICO@HOTMAIL.COM - 2006

dos del servicio Web, lo que facilita conaiderablemente lu c:osu. De est.. forma podemos interactuarcon el servicio Web como si se tratara de una clase PHP local..

A continuad6n, definimos una matriz de los parnmetros que se deben pasar a la consulta browaenode:

```
tpar   ter•('miode'l•ttlode'
fpar   t•r•['page')•$page,
fpar   t•r•I'type'l•'b.e8vy',
$par   t•r•('tag'l•ftbi•'>_  ...oclD'
$par   t•r•l'devtag')•ftbi•'>_d8...rmg,
fpar   ter•l'•ort'l•'•••l••raak'l
fpara—ten['browH_DOCle')  • fb-llocle,
```

Por medio de la clue proxy, invocamos los m6todos del servido Web y puamos la matriz de parúnetros:

Desarrollo Web con PHP y MySQL

Los datos abnac:enados en $reaul t forman una matriz que podemos almacenar directamente en nuestra matriz _produc:ta en la clase Amazonlleaul tSet ..

Almacenar los datos en caché

Volvamos a la funci6n getARS () para analizar el almacienamiento en cac:hf. Como recordañ, la ÍlllleiéJ\ tiene este upecto:

```
-¡/ Obtenga IIII claH  AmaaaabnltSet de la cacb6 o a tra'ri• da WUI -nlta
// Si opta por IIIIt C0118Ulta,  all&dala a la cacb6.
f1111ction gel:ARS(ftype, fpar...ter•I
(
   $cacbe  • cacbed($type,  $par...ter•I,
   if($cacbel  // •i exi•t• en cacb6
   (
      return   $cacbe,

   el••                                                    '''
   (
      S•r•  •...., AmaaonllenltSet'
      if1$type • •  '••in'I
         $ar•'>ASDISearcb(padASl•<Spar     ter•l'••in'll,
                                  $par    ter•  l'IIOde•l1,
      if($type • •  'browse'I
         $ar•'>brow8..ode8earcb(Spar   t•r•['brow•enode'),
                                 $par   t•r•['page'),
                                 Spar   ter•['IIDde')l1
      if (Stype • •  'H&rCb')
         S•r•'>kllywordSearcb($par ...ter•l'•••rcb'J,    $par...ter•['page'),
                                 $parameter•l'IIDde,Jl1
      cacbe(Stype,  $par...ter•,  $ar•I,

   nturn   $ar•,
```


Todo el almacenamiento en c:achf SOAP o XML de la aplicadón se realiza a traws de esta función.

o.ponemos de otra para almacenar ilMgenes en caché. En primer lugar, invocamos la función c:ac:bed () para comprobar si el AllazonReaultset ya estj en cac:M. Si kJ est4, devolvemos los datos en lupr de realizar una nueva solicitud a Amuon:

```
$cacbe • cacbediStype, $par...t•r•I,
if(tcacbel  // •l ••i•t• en cacb6
(
   return   $cacbe1
```

En caso contrario, al obtener los datos de Amazon, los aftadimos a la cachr.

Veamos estas dos funciones: c:ac:bed() y c:ac:be ( ), que mostramos en el listado 33.12. Estas funciones implementan el almacenamiento en caché que Amazon requiere como parte de sus condiciones.

u.tallo n.12. Funcionescache() y C8dled(). Funcionesde almacen81111erii:1 en cac:116 de c:echelunc:llan.pllp.

```
// cmpruebe •i l-  dat-  da a.a-   Htta  en cacb6
// al  •tta,  daw6lvalo•
// en ca•o —t.rario,  dawuelva fal••
functi-  cacbed<Stype, $par...tenl
(
   if($type  •  •  •brow••'I
   (
      $filena.e  •  CAC'JIS.'/braw••'•$par    ter•l'brow•enode').•.
            '.$par ...ter•[•page•).•.$par       ter•['IIOde').•.dat'l

   if($type  •  •  '..arcb')
   (
      $filena.e.  CACBS.•/  ..arcb.•.$par    tera('•••rcb').•.
            •.$par ...ter•[•page•).•.fpar       ter•[madaJ..dat'l

   ifC$type  •••'  ..in')
   (
      $fileDU18.  CACBS.'/a•in.•.$par  ...t•r•l'••in•J.•.$par   ...ter•
('IIDde'). '.dat'l
   }

   // ¿faltan lo• dato• en cacb6 o ti-      8'•  de  1 bora?
   if(lfile exi•t•(Sfilena.el  ||
         IIIIIIl:ti .. ll  •  fileati .. 1$fileDU18II  > 10•10II
   (
      return ral .. ,
   }
   $data•  file..9•t_-tent•l$filaDU1el1
   return un..rialiae($datal,
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
// allada lo• dato• de Amazon a la cacbt
function cacb8($type, Spara..ter•. $datal
(
   if($type • • •brov•e'I
   (
      $filena .. • CACRll.'/brow•e.'.$para  ..ter•['brov8enode'I .'.'
            .$para ..ter•[•page•J .'.•.$para  ..ter•l'mode'l.'.dat';
   ▌
   if1$type • • ••earcb'l
   (
      $filen-   • CACll&.'/•earcb. ' .$par — ter • ['Harcb•J".' .'
            .$para ..ter•l'page•J .'.•.spara  ..ter•['mode').•.dat';
   ▌
   if 1$type • • •aain' 1
   (
      $fil..... • CACBB.'/a•in.'.$para   ..tera[•aain'J.•.•Spar-ter•
1'-«le' I. ' .dat' 1
   ⎿
   Sdata • ••rializa($datal;

   $fp • fopen1$filename,   'wb'l1
   ifl1$fpl1(fwrit•C$fp,   $date1•  • ·11  1
   (
      echo  (•cp>lrror, could not •tore cacb8 filec/p>'l1
   ▌
   fcloae(Upl1
```

Al analizar este código, comprobad que los archivos en caché se almacenan con un nombre de archivo formado por el tipo de la cONUltaseguido por los padmetros de la misma. La función cache <l serializa los resultados para almacenarlos y la función cached ( ) los deserializa. Ésta última también sobrescribe todos los datos que tengan m,s de una hora, como se indica en las condiciones. La función aerialize (l convierte datos de programa almacenados en una cadena que se puede almacenar. En este caso, creamos una representación de un objeto AmazonResultSet que se pueda almacenar. La invocación de unaerialize (l se encarp de lo contrario, es decir, convierte la versión almacenada en una estructura de datos en memoria. Debe saber que al deserializar un objeto como éste, es necesario incluir la definición de la clase en el archivo para que la clase sea comprensible y utilizable después de cargarla otra vez. En nuestra aplicación, la recuperación de un conjunto de resultados de la cachf se realiza en una fracción de segundo. La ejecución directa de una nueva consulta puede tardar hasta 10 segundos.

Crear el carro de la compra

Una vez descritas todas estas opciones de consulta de Amazon. lo más obvio que podemos hacer es aear un carro de la compra. Como ya hemos deacrito ampliamente esta operación en cap(tulos anteriores, no la analizaremos en este apartado. Las funciones del carro de la compra se muestran en el listado 33.13.

U.... 33.13. c:artfuncllons.php. lmplernent8ción del carro de la compra.

```
c?pbp
require_once('AmazonRe•ultSet.pbp');

// Al utUiur la funci6n •howS-ryO  del archivo bookcliaplay.pbp ••
// ...•tran loa contenido• actual•• del carro de la c0111Pra
function •llowCartl$cart, $-«le)
(
   // genere la ..tris que •• va a paaar
   sproduct• • arrayll1
   foreacbl$cart aa $ASI••>$product)
   (
      Sara • getUS ('aain•, array('aain'•>$ASI•,  'moda• •>s-del1;
      ifl$anl
         Sproducta[J • S•r•->getProductlOl1
   ▌
   // cree el foraulario que ae ve a vincular al carro de la ccmpra de Amazon.c-
   acho •cforia Mtbod••POST•
         action••bttp://.....,.a..zon.coa/o/dt/a••oc/handle-buy·box•>•;
   foreacb($cart a• $ASI••>$product1
   (
      $qllantity • $cert[SASI•JC•quantity'J;
      echo •cinput type•'bidden•  na..••aain.$ASI•'  velu••'$cau-Dtity'>°1
   ⎿
   echo 'clnput type••bidden• aa..••tag-value• value••ASSOCIATSID">';
   echo •cinput typ•••hidden• na..••tag_value• value.•ASSOCIATSID°>';
   echo 'cinput type••i..ge• arc••imagea/cb8ckout.gif•
                           n&M•"•ub1lit.add-to-cart•
                           valu••"1luy  ro11 Amazon.cama">';
   echo ' Wben you bave Uniab8d abopping pre•• cb8ckout to add all ti.
      it-  in your Tabuayo cart to yqur —•on cart and C0111Pl•t•
      your purc:haae.cbr  />';
   echo •c/fora>';

   echo •ca bref • •iadex.pbp?..:tion•..Ptycart•>ci11g
         •re•  •iaagea/emptycart.gif"  alt • •lmpty Cart• border • O>c/a>
      If you b8ve finiabed witb thia cart, you can e11pty it of all it•••·
      cbr />' 1
   echo 'chl>Cart Cantlll1t•c/bl>';
   allov8-ry($product•, 1, c-t ($produc:tal, $110de, O, true):


/I maeatre el re•-n del carro que siel1pre aparece en pantalla
// mueatre Gnica11ente lo• tre• dltimoa articulo• allaclidoa
functian •bowlmallC&rt()
(
   global S_SISSIOII;

   echo •ctable border • 1 cellpadding • 1 cell9pac;1ng • 0>'1
   echo •ctr>ctd claa• • cartbeading>rour Cart $'.nllll1ber_format
(cartPrice(I,  21.
      'c/td>c/tr>'.
   echo •ctr>ctd clá•• • cart>'.cartContenta().'c/t:4>c/tr>'1
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
   // fozmalario para vincular al carzo de la compra de Allason.coa
   acho 'cfora Mthod•"IOST"
        action•"http://1111W ....son.com/o/dt/a••oc/halldle·buy·box•>'1
   acho •ctr>ctd el••••   carthaadi119>ca href •
                            "illdes:.php?l!etiOD••llowcart•>ci119
                            •rc•"i•ge•/detail•.g:lf• border•O>c/a>',
   foraach1$_8a8IOll['cart•J a• $Mill•>Sproduct)
   {
      $quantity • $_sa&IOll['cart•J[$MIIII ['quantity•J1
      echo •ciaput type-'hiddea' ......,'aeia.$MI••    valua•'$quantity'>"1

   acho •ciaput type•"hiddea• aa.. ••ta9-valua• valua-"A880CIATSID">'1  '
   acho •ciaput type•"hiddea"  - ··ta,_value•    valua•"ABSOC:IATSID">' ,
   acho 'ciaput type•"t. ..,-• •rc••i.-.re•/chackollt.9if"
        a.1111••111lbiait.add·to-cart• valua•"Buy Proa Allasoa.coa•>';
   echo •c/td>c/tr>'1
   echo 'c/fora>';

   acho •e/tabla>',


// -acre lo• trH llltimo• articulo• dadido•  al carzo
fuaction cartCoatenta(I
{
   9Íobal S_1888IOll1

   $display•  array_•lice($_1&88IOll['cart'J,  -3,  31,
   .JI. fIll•$•-•qu•  IIP1'1'91!CIID  a olNlea cr4?11016gico iaveno
   $di•play • array_revereelfdiaplay, true),

   $neult  • '',
   Scounter • o,

   // ai loa nombre• -  utanao•, redGscalo•
   foraachlSdi•play a• Sproductl
   {
      ifC•trlaalSproduct[•-•J)c•IO)
         $ra•ult '•  $product[•-·J.'cbr  />';
      al••
         Sraeult '•  au!»•tr(Sproduct[•a,1119•),  0,  37).'  ''' cbr  7>'1
      $c011ntar++1


   Il a&ada Una••  ea blanco •i al car:ro ••tA prActic-ta vaclo para coa•arvar
   // eu rapra•aatacian
   forC,$countarc3l  $couatar++I
   {
      $raeult .•  'cbr  />'1
   }
   return Sra•ult,


// calcula al prac:lo total de lo• articulo• del carro
funct:loa cartPricaÇI
{
   9lobal $_1U8IOII;
```

```
   Stotal • o.o,
   forNchC$ _IS18IOll['cut'I •u Sproduct)
   {
      Sprice • atr_rapl-1'  $',  '',  Sproduct ['prica•J1,
      $total+•  Spr:lca•Spzoduct[•quant:lty'l1


   return $total,


// dada  un articalo al carzo
// actualMnta  no M pIIÑ8  dad:lr •• de uno por vas
fuact:lon mdd'roeart(..cart,  $MI9,  $modal
{
   ifli••atCScart[$Miml  11
   {
      Scart[$MDI  l''lllUtity'I  +•11
   }
   al••
   {
      // comprueba  qu  al ASD •  vAlido y buqu  al precio
      .. re•  -  Mall0deaultht1
      Sproduct • $an·>Mill8Nrchl$MI•,  $moda)1

      if($produc:t•>Yal:f.d())
         $cut[$MIIII • array( •prica°->Sproduct·>OUrPriceCI,
               ........ •> Sproduc:t•>product•aMII,  •quaat:lty•  •> 1)  ,



// alill:laa al articalo del car:ro
function delatarra.cart(.SC:art,  $MIIII
{

   llDNt Cfcart [$MIIII 1 1
}

? >
```

No obatante, hay .igunu diferenciaa con respecto a este cano de la aapra, por ejemplo en la fund6n addToCart (›• Al tratar de aftadir un artfc:ulo .i cano, podemos comprobar que es un ASIN rilido y buscar el precio actual (o .i menos el .Imacenado en cachf).

La pregunta mú inlelellanle es cómo enviar los dates a Amaon cuando los clientes terminan.

Conectar a Amazon

Ff,ae en la fund6n sbowCart (› del listado 33.13. Veamos la parle princip-1 de la misma:

```
// craa al fozmalario para v:lacular a un car:ro ele la C0119ra de Mason.coa
echo 'cfon  • thod•"IOST"
```

EL_HOLISTICO@HOTMAIL.COM - 2006

```
      actiOD•"http1//....., .... zon.c011/o/dt/ae110C/baadle-buy-box•>'1
foreach(Scart •• SASI••>$productl
(
   $qualltity • $cart($ASI•J(•quaatity'I;
   echo •cinput type••'hidden• n...•'aain.$ASIR'   value•'$quantitY'>";

echo •cinput type•"hidden• na-··ta9-value•  value•"ASSOCIATBID">';
echo •cinput type•"hidd9n• na-··•ta9_valua• value•"ASSOCIATBID">';
ecbo •cinput type••iuga•  erc••i ..gea/cbackout.gif•
      na..••eut.it.add-to-cart• valua••&uy Proa Amazoa.coa">'1
echo•  Wb8a you llave finiahed ebopping pre•• checkout to add all the it..•
   in your Tahuayo cart to your Amazon cart and complete your purchaee.
   cbr />
echo 'e/fon>';
```

El botón Checkout es un botón de formulario que conecta el carro a un carro de la compra de Amazon. Enviamos diferentes ASIN, cantidades y nuestro Id. de asociado a través de un variable POST. El resultado final que se obtiene al pulsar este botón se muestra en la figura 33.5, al principio del capítulo.

La ¡,'gina resultante tendrá un aspecto diferente si el usuario no tiene almacenada una cookie de Amazon en su equipo para que Amazon lo identifique, pero el resultado final será el mismo.

Una de las dificultades de esta interfaz es que sólo funciona en una dirección. Podemos añadir artículos al carro de Amazon, pero no eliminarlos. Esto significa que los usuarios no pueden pasar de un sitio a otro sin terminar con artículos duplicados en sus carros.

Instalar el código del proyecto

Si quiere instalar el código del proyecto correspondiente a este capítulo, tendr4 que realizar una serie de pasos poco habituales. Una vez ubicado el código correctamente en su servidor, tendr6 que hacer lo siguiente:

- Crear un directorio caché.
- Definir los permisos en el directorio cac:hé para que las secuencias de comandos puedan esaibir en el mismo.
- Modificar constante. php para que indique la ubicación de la caché.
- Solicitar una etiqueta de programador de Amazon.
- Modificar constante. php para que incluya esta etiqueta y, opcionalmente, su Id. de asociado.
- Comprobar que NuSOAP esti instalado. En nuestro caso se incluye en el directorio Tahuayo, pero puede cambiarlo de sitio y modificar el código.
- Comprobar que ha compilado PHP con compatibilidad con XML

Ampliar el proyecto

Puede ampliar este proyecto de formas muy divertidas:

- Puede ampliar los tipos de búsquedas disponibles a través de Tahuayo.
- Amazon también dispone de un servicio Web XSLT con el que puede probar.
- En los servicios Web de Amazon se incluyen enlaces a innovadoras aplicaciones de ejemplo, como es el caso de http://associates.amazon.com/exec/panama/aeeociatee/ntg/browse/-/567634.

Un carro de la compra es el elemento mú obvio que se puede crear con estos datos, pero no el único.

Lecturas adicionales

Existen millones de libros y recursos en linea sobre los temas de)CML y servicios Web. Un punto de partida muy acoosejable es W3C, en la página del Grupo de trabajo XML http: / /-w. w3. org/XML/Core y en la página de servicios Web, http://~.w3.org/2002/ws.

EL_HOLISTICO@HOTMAIL.COM - 2006

$name();

Apéndices

Apéndice A
Instalar PHP y MySQL

Existen veniones de Apache, PHP y MySQL disponibles para distintas combinadones de sistemas operativos y servidores Web. En este apéndice describiremos cómo puede configurar Apache, PHP y MySQL en diferentes plataformas de servídor. Analizaremos las opciones más comunes disponibles pan Unix y Windows XP. Entre los aspectos que analizaremos destacamos los siguientes:

- Ejecutar PHP como intc!rprete CGI o como módulo
- Instalar Apache, SSL, PHP y MySQL en Unix
- Instalar Apache, SSL, PHP y MySQL en Windows
- Comprobar el funcionamiento: phpinfo ()
- Añadir PHP y MySQL a Microsoft Internet Information Server
- Instalar PEAR
- Otras posibles configunciones

El objetivo de este apéndice es ofrecerle una guía de instalación de un servidor Web que le permita alojar varios sitios Web. Algunos sitios, como en los ejemplos que hemos descrito, requieren SSL para soluciones de comercio electrónico. Y la mayoría se controlan por medio de secuencias de comandos para realizar conexiones a un servidor de base de datos (08) y extraer y procesar datos.

Para muchos usuarios de PHP nunca será necesario instalar PHP en un equipo, razón por la que este material se incluye en un apéndice y no en el primer capítulo.

EL_HOLISTICO@HOTMAIL.COM - 2006

La forma mú séncilla de acceder a un servidor fiable con una conexión rápida a Internet y PHP instalado consiste en contratar una cuenta de uno de los miles de servicios de alojamiento existentes.

En función ~e la: ~ón por la que Instale PHP en un equipo, puede tomar decisiones diferentes. Si dispone de un equipo conectado permanentemente a la red que tiene pen~do utilizar como servidor, el rendimiento le resultar6 de gran importancia. Si desea crear un servidor de desarrollo en el que generar y probar su código, el hecho de disponer de una configuración similar al servidor activo ser6 la consideración m6s importante. Si pretende ejecutar ASP y PHP en el mismo equipo, ~dr6 que aplicar otras limitaciones.

Ejecutar PHP como intérprete CGI o como módulo

El intfrprete de PHP se puede ejecutar bien como módulo o como binario CGI (Interfaz de pasarela com11n) independiente. Por lo general, la versión modular se utiliza por motivos de rendimiento. Sin embargo, se suele utilizar la versión CGI en servidores en los que la versión modular no est6 disponible o porque permite a los usuarios de Apache ejecutar distintas p6ginas en las que se ha activado PHP bajo diferentes Id. de usuario.

En este apéndice nos centraremos principalmente en la opción modular como método de ejecución de PHP.

Instalar Apache, PHP y MySQL en Unix

En función de sus necesidades y de su nivel de experiencia con sistemas Unix, puede optar por realizar una instalación binaria o por compilar los programas directamente desde el código fuente. Ambos enfoques tienen sus ventajas.

Para un experto, una Instalación binaria sólo le In!\rar6 VariCJ!l minutos y a un principiante no mucho mú, pero resultad en un sistema que se encuentra una o dos versiones por detrú de las actuales y que estar6 configurado con las opciones definidas por alguien mú.

Una instalación de código fuente tarda varias horas en descargarse, instalarse y configurarse, y la operación resulta un tanto intimidatoria las primeras veces. Sin embargo, le permite tener un control total. Seleccione lo que quiere instalar, qué versiones utilizar y qué directivas de configuración definir.

Instalación binaria

La mayor parte de las distribuciones de Linux incluyen un servidor Web Apache preconfigurado que incorp9ra PHP. Los detalles de lo que ofrece dependen de la distribución y la versión seleccionadas.

Un inconveniente de las instaladoms binarias es que raramente se obtiene la 41tima versión del programa. En"fanci6n de la importancia que tengan lu 41timas versiones de los parches, puede que en su caso no sea importante. El mayor problema es que no se pueden se1ecc:ionarlas opciones compiladas en los programas.

La solución mú flexible y fiable consiste en compilar todos los programas-que necesite desde sus códigos fuente. lleva mú tiempo que instalar los RPM, por lo que puede optar por utilizar los RPM u otros paquetes binarios disponibles. Incluso si los archivos binarios no estm disponibles en fuentes oficiales con la configuración que necesita, sin duda encontrar6 otros no oficiales con ayuda de un motor de búsqueda.

Instalación desde código fuente

A continuación Instalaremos Apache, PHP y MySQL en un entorno Unix. En primer lugar, será necesario decidir qué módulos adicionales cargaremos. Como en algunos de los ejemplos descritos en el libro se utiliza un servidor seguro para transacciones Web, instalaremos un servidor en el que se haya activado SSL.

Nuestra configuración PHP coincidirá aproximadamente con la configuraaón predeterminada pero también veremos cómo activar las dos siguientes bibliotecas en PHP:

- gd2
- PDFlib

Son dos de las muchas bibliotecas disponibles para PHP. Las incl~ para ,ue se haga una idea de cómo se habilitan bibliotecas adicionales en PHP. La compilación de la mayoria de los programas de Unix es similar.

Habitualmente es necesario compilar PHP después de instalar una nueva bib~oteca, por lo que si sabe lo que a va necesitar con antelación, instale en su equipo todas las bibliotecas necesarias y, tras ello, compile el módulo PHP.

Realizaremos la instalaci6n en un servidor SuSE Linux, pero es lo suficientemente genérica como para aplicarla a otros servidores Unix.

En primer lugar recopilamos los archivos necesarios para nuestra instalaaón, para lo que necesitaremos los siguientes programas:

- Apache (bt tp: / /bttpd. apache. org): El servidor Web
- OpenSSL (bt tp: / /www.opensal . org): Conjunto de herramientas de código abierto que implementa SSL
- Mod_SSL (bttp://www.modsd.org): Proporciona una interfaz modular de Apache a OpenSSL
- MySQL (bttp: / /www:mysql. com): La base de datos relacional
- PHP (bt tp: / /www.pbp, net): Lenguaje de secuencia de comandos del lado del servidor

EL_HOLISTICO@HOTMAIL.COM - 2006

- http://www.pdflib.com/producta/pdflib/download/index.html: Biblioteca para ge~erar documentos PDF sobre la marcha
- ftp: //ftp. uu .net/graphics/jpeg: La biblioteca JPEG, necesaria para PDFlibygd
- http://www.libpng.org/pub/png/libpng.html: La biblioteca PNG, necesaria para gd
- http://www. gzip .org/zlib: La biblioteca zlib, necesarias para la biblioteca PNG anterior
- http: //-w. libtiff .org: La biblioteca TIIF, necesaria para PDFlib
- ftp: / / ftp. cae. washington. edu/ imap: El diente c IMAP, necesario para IMAP

Si desea utilizar la función mail (), necesitará instalar un MTA (Agente de transferencia de correo), aunque en este apéndice no analizaremos dicho proceso.

Suponemos que dispone de acceso como usuario raíz al servidor y que ha instalado las sigui~tes herramientas en su sistema:

- gzip o'gunzip
- gcc y GNU make

Cuando esté listo para iniciar el proceso de instalación, primero debe descargar todas las fuentes de archivos tar a un directorio temporal. Asegúrese de que díspone de sufiqente espacio para ello. En nuestro caso, hemos seleccionado /usr / src como directorio temporal. Debería descargar estos archivos como usuario raíz para evitar problemas de permisos.

Instalar MySQL

Vamos a realizar una instalación binaria de MySQL. De esta forma, los archivos se ubican automáticamente en distintos puntos. Los directorios que hemos seleccionado para el resto de los programas son los siguientes:

- /usr/local/apache
- /usr/local/ssl

Puede instalar la aplicación en directorios diferentes con tan sólo cambiar el prefijo antes de realizar la instalación.

Manos a la obra. En primer lugar, conviértase en usuario rafz por medio de su.

```
1 •u root
```

e introduzca la contraseña del usuario rafz. Cambie al directorio en el que haya almacenado los archivos fuente, por ejemplo

```
l cd /u•r/•rc
```


MySQL recomienda actualmente que los usuarios descarguen un binario de MySQLen lugar de compilar desde cero. La versión que utilice depende de lo que quiera hacer. Al cierre de esta edición, la versión de producción de MySQLen la 4.0. Necesitará la ·U si desea utilizar subconsultas y la 5.0 si desea utilizar procedimientos almacenados. Cuando lea este libro, puede que estas versiones se hayan convertido en la de producción. Aunque las versiones iniciales de MySQLsuelen ser muy estables, no es aconsejable utilizarlas en un sitio de producción. No obstante, para aprender y experimentar puede que le sirvan.

Debería descargar los siguientes paquetes:

```
MySQL·•erver·VE'RSIÓN.1386.rpa
MySOL-Max-VERSIÓN.1386.rpm
MySOL-client-VERSI6N.i386.rpm
```

(La palabra *VERSIÓN* es un marcador de posición correspondiente al número de versión. En función de la versión que seleccione, asegúrese de que escoge el correspondiente conjunto.) Si tiene pensado ejecutar el cliente y el servidor MySQL en este equipo, y compilar compatibilidad con MySQLen otros programas como PHP, necesitará todos estos paquetes.

Introduzca los siguientes comandos para instalar los servidores y el cliente MySQL:

```
rpm -i MySOL·••rver-VE'RSIÓN.1386.rpa
rpm ·i MySQL·Max•VE'RSIÓN.1386.rpm
rpm ·I MySOL·cl!ent·VERSIÓN.1386.rpm
```

Con esto instalará y podrá ejecutar el servidor MySQL

Ha llegado el momento de asignar una contraseña al usuario raíz. En el siguiente comando, no olvide sustituir new-password por una contraseña propia; en caso contrario, new-paaaword será su contraseña raíz:

```
my•qladlllin -u reot pa••word •new-pa••word'
```

Cuando instale MySQL,creará dos bases de datos de forma automática. Una es la tabla msyql, que controla permisos de usuarios, host y bases de datos en el servidor. La otra es una base de datos de prueba. Con la siguiente línea de comandos podrá comprobar el sistema:

```
# ay•ql -u root -p
Enter pae•word:
mymql> •how dataha•e•;
+--------------------+
| Databan            |
+------------------·--+
| mymql              |
| teec               |
+---·----------------+
2 row• in ••e (0.00 •ecl
```

Escriba qui t o \q para cerrar el cliente MySQL

EL_HOLISTICO@HOTMAIL.COM - 2006

La configuraci6n predetemünacip de MySQL permite a cualquier usuario acceder al sistema sin necesidad de proporcionar un nombre de usuario y una contrase,. fta,. ló que evidentemente no es acomejable.

La \iltima parte obligatoria de M)ISQLaonsiste en eliminar el usuario anónimo. Para ello basta C0ñ abrir una línea de comandos y escribir lo siguiente:

```
, myaql -u root -p
my•ql> u•• Íly•ql
ay•ql> clelete froa ueer wber• Ueer•''1
•yeqb  quit
```

Tras ello, escriba

```
my•qledain -u root -p reload
```

para aplicar los cambios.

También debería habilitar el registro binario en su servidor MySQL ya que lo necesitad si tiene pensado utilizar la replicaci6n. Para ello, primero debe detener el servidor:

```
ay•qladain -u root -p •butdown
```

Cree un archivo con el nombre /etc/my. cnf que puede utilizar como archivo de opciones MySQL Por el momento, sólo necesita una opci6n, aunque puede contar con varias. En el manual de MySQL encontrad la lista completa.

Abra el archivo e introduzca

```
lmy•qld)
log-bin
```

Guarde el archivo y sslga. Reinicie el servidor mediante la ejecución de myaqld_safe.

Instalar PDFlib

Si no quiere utilizar PDFlib para crear archivos PDF como vqnos en un capítulo anterior, puede ssltarse este apartado. Descargue PDFLib de ht tp: //wwv. pdflib.com/producta/pdflib/download/index.html.

Para extraer los contenidos del archivo PDFlib, escriba:

```
I gunsip -e pdflib-4,0.3,tar.gs | tar xvf -
```

No utilizaremos PDFhb directamente, por lo que volveremos a este punto una vez se ejecute PHP.

lnslar PHP

. Debería co~tinuar como usuario r~; en caso contrario, utilice su para convertirse en usuano rafz. Antes de poder instalar PHP, es necesario preconfigurar Apache para que sepa dónde se encuentra todo. Volveremos a este punto ms adelante

cuando configuremos el servidor Apache. Cambie al directorio en el que tenga los archivos fuente.

```
I cd /u.r/.rc
I gunsip -e apacbe_l.3.27.tar.gs | tar XYf -
I cd apacbe_l.3.27
1 ./configure - ·prefix•/uer/local/apacbe
```

Ya puede empezar a configurar PHP. Extraiga los archivos fuente y dmbielos a su directorio:

```
I cd /u.r/.rc
I gunsip -e php-4.2.3,tar.gs  | tar XYf -
I  cd pbp-4.:.3
```

Como en el caso anterior, existen numerosas opciones para el comando conf igure de PHP. Utilice ./configure --help | leHparadeterminarloquequiere alladir. En este caso, queremos alladir compatibilidad con MySQL,Apache, PDFlib ygd.

Todo lo que viene a continuación se incluye en un solo comando. Podemos alladirlo todo en una lfnea o, como hemos hecho aquf, utilizar el caricter de continuación, la barra invertida (\), para poder escribir un comando en varias Uneas para mejorar la legibilidad.

```
•• /configure --witb-my11qli-.y9ql_config_patb/ay11ql_config\
              --witb•apacbe• ••/apacbe_l,3,31 \
              --wltb-jpeg-dlr•/ruta/a7jpaglib \
              --witb·tlff·dir•/ruta/a.tiffdir \
              --witb-slib·dir•/ruta/a/slib  \
              ··witb·i..pa/ruta/a/1.-pcclient \
              - ·witb·gd
```

Tras ello, instalamos los binarios:

Copiamos un archivo i~i en el directorio l ib:

```
1 c:p plip.1ni'~i1lft  ,u1rllocclfttbfpbp.ini
```

o

```
I cp pbp.ini-reCC111111encled  /u•r/local/lib/pbp.lni
```

Las dos versiones de php • ini de los comandos sugeridos tienen un conjunto de opciones diferente. El primero, php . ini-diat, esti ideado para equipos de dessrrollo. Por ejemplo, configura display _errors con el valor On. De esta forma se fadlita el desarrollo, pero no resulta adecuado para un equipo de producdón. Cuando en el hbro hemos hecho referencia al valor predeterminado de la configurad6n de php. ini, nos referiamos a su configuración en esta versión de php. ini. La segunda versión, php . ini-recommended, esti indicada para equipos de produc-

EL_HOLISTICO@HOTMAIL.COM - 2006

á6n. Puede modificar el an:hivo php. ini para definir las opcioNs de PHP. Existen distintas opáones que puede configurar, pero conviene destacar algunas en concreto. Puede que tenga que establecer el valor de •endmail_path si quiere enviar correo electrónico por medio de secuencias de commdos. A continuadón, configuramo11 OpenSSL, que utilizaremos para crear certificados temporales y archivos CSR.. - -pref ix especifica el directorio de instalación principal.

```
I gunsip -e op.... •1·0.9.&g.tar.gs  | tar xvf •
I cd op912H1•0.9.6g
1 ./coafig  --prefix•/u•r/locall••l
```

Tras ello, lo creamos, lo probamos y lo instalamos:

```
' •k•
' •k• t••t
1 •k• ianall
```

Configuraremos el módulo mod_SSL y~ tras ello, lo especificaremos como módulo que se pueda cargu con la configuradón de Apache.

```
I cd /uer/erc/
I guasip -ci 90Cl_H1·2.1.11•1.3.2'7.tar.gs     |tar xvf
I cd mod ••1·2,8.1I·1.3.2'7
1 ./coafigi,rtt  --w~tb•apache• " /apacbe_1.3.2'7
```

' Ffjese en gue podemos añadir más módulos de Apache el úbol fuente. La opción --enable-•hared•••l permite generar mdo_SSLcomo libHl. se OSO (Objeto dinúúco computido). Consulte la documentaá6n INSTALLhtdoc•/manual/d•o-luatl del 4rbol fuente de Apache si necesita más informad6n sobre compatibilidad OSO en Apache. Bs recomendable que los ISP y los que mantengan paquetes utilicen la opción OSO para obtener una nwcima flexibilidad con mod_SSL Sin embargo, Apache ~o es compatible con OSO en todas las plataformas.

```
I cd ••/apacbe~1.3.2'7
I  SSL IIAS1•.• /opeaeal-0.9.6g   \
      - ./configure \
        --eaaltla·IIOdula•••l  \
        --activate•IIOdule•arc/90Clulea/pbp4/libpbp4.a   \
        --prefix•/uar/local/apacba  \
        --aaaltla·abered•••l
```

Por último, puede utilizar make con Apache y los certificados, e instalarlos.

```
' •k•
```

Si ha realizado correctamente todos los pasoa, obtendn un memaje similar al siguiente:

```
'-------------------------------------------------------------------'
  ..fore yi,u iaatall tbe package you DOW ebould prepare tbe SSL
| certif.cate •v-tea oy rlllllliJl9 tbe 'llake certificate' cGe11U1d.
| ror differeitt aituatioa• tbe followiJl9 variante are provided:
```



```
I uke  certificate TYPl•dwlmy    (dullay ••lf-aignad Saake Oil certl
I make certificate TYPl•teet     Cteat cert aigaed by &aake Oil CAi
I make certificate TYPl•cuatoa   (cuatoa cert eigaed by owa CAi
I malee certificate TYPl•exiatiag Cuiatiag cert)
       CRT•/patb/to/your.crt   [IIY•/patb/to/your.key)

uae TYneduaay    wbea you'ra a  veador package -iataiaer,
tbe TYPleteat    wbea you're aa acllia but waat to do teata oaly,
tbe TYPl•cuatoa  when you•re aa adaia williag to rua a real aerver
aad TYPl•exiatiag wbea you•r• aa adaia wbo upgradea a aerver.
(Tbe default ia TYPl•teat)

Additionally add ALGO•ll8A  (default)  or ALGO-DSA to ••lect
tbe •iga&ture algoritlll Ulled for tbe geaerated certificate '

Uae 'llake certifi~ate VIIIf•l'  to diaplay tbe geaerated deta.
```

```
                                   Ralf s. lagelacball
                                   ra..eagalecball.com
                                   www.eagalacball.coa
'-------------------------------------------------------------------'  ----+
```

Ya puede crear un certificado personalizado. Bsta opción requiere una ubicación, una compañia y algunos otros elementos. En cuanto a la informaá6n de contacto, conviene utilizar datos reales. En las preguntas restantes del proceso, basta con la respuesta predeterminada.

```
I malee certificate TYPl•cuetoa
```

Seguidamente, instale Apache:

```
II mak• iaetall
```

Si todo sale correctamente, ven un mensaje como •te:

```
'-------------------------------------------------------------------'
  You aow have eucceaafully built aad iaetallad tbe
  Apache 1. 3 BTTP Hrver. To verify tbet Apacbe actually
  worka correctly you now •bould f!rat cbeck tba
  (iaitially created or preeenredl config,aration file•

    /uar/local/apacbe/conf/bttpd.conf

  aad thea you ahould be able to 1-.diately fire up
  Apacbe tbe firat ti•  by ruaaiag:

    /uar/Local/apacbe/bia/apacbectl etart

  Or wbea you waat to rua it witbssL eaablad uee:

    /Ullr/local/apacbe/bia/epacbectl etartaal

  Tbaaka for ueing Apacbe.        ~ · Apacbe Group
                                  bttp://www.apacbe.org/
'-------------------------------------------------------------------'
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Puede comprobar si Ápache y PHP funcionan. Sin embargo, tendr6 que modificar bt tpd. conf para aftadir el tipo PHP a la configuración.

Fragmentos de código de httpd-conf

Fíjese en bt tpd. conf y at\ada las siguientes lfneu. Si ha seguido lu ~trucciones anteriores, su archivo bttpd. conf se enc:ontrar6 en el directorio /usr/local/apacbe/conf. El archivo inc:luyeel addtype para PHP con comentarios. Debería anular el comentario para que su upec:tosea el siguiente:

```
Adch'ype application/x·httpd·php .php
AddType application/x·httpd·php-aourca .phpa
```

Ya puede iniciar el servidor Apache para comprobar si funciona. En primer lugar, iniciaremos el servidor sin compatibilidad con SSLpara ver si aparece. Comprobamos dicha compatibilidad y, tras ello, detenemos el servidor y lo iniciamos con la compatibilidad SSL activada para comprobar que todo funciona correctamente.

configtest comprueba si la configuración se ha realizado correctamente:

```
t cd /uer/local/apacha/bin
1 ./apachectl configt••t
Syntax Ólt
1 ./apachectl •tart
  Japachectl nart, httpd nartad
```

Si ha funcionado correctamente, cuando nos conectemos al servidor con un navegador Web, el resultado ser6 similar al de la figura A.1.

No se puede conectar al servidor con un nombre de dominio o por medio de la dirección IP del oi:denador. Compruebe ambos calOS para asegurarse de que todo funciona correctamente.

ifunciona la compatibilidad con PHP?

A continuación comprobaremos la compatibilidad con PHP. Cree un archivo con el nombre test. pbp en el que debe incluir el siguiente c:ódigo. El archivo debe estar en la ruta raíz del documento que, de forma predeterminada, ser6 /uar / local/apacbe/btdoca, aunque depende del prefijo que haya seleccionado inicialmente. Sin embargo, se puede cambiar en ht tpd. corif.

```
c?php phpinfo();  ?>
```

La pantalla de resultado ser6 similar~ la mostrada en la figura A.2.

Figura A.1. pjgln• de pruebll predelennlMda que propon:10118 Apache.

Figura A.2. La función phplnfo() proporcionll información de configuración muy lillt.

EL_HOLISTICO@HOTMAIL.COM - 2006

!Funciona SSL?

Ya podemos pasar a probar SSL. En primer lugar, detenga el servidor y reinfcielo con la opción SSL activada:

```
# /uar/local/apache/bin/apacbeccl •top
1 /u•r/local/apache/bin/apachectl•tart••l
```

Compruebe si funciona. Para ello debe conectarse al servidor con un navegador Web y seleccionar el protocolo **http**a como se indica a continuación:

```
http•://•u-•ervidor.•u_dollinio.cOII
```

Pruebe con la dirección IP de su servidor:

```
http•://xxx.xxx.xxx.xxx
```

o

```
http://xxx.xxx.xxa.xxx:443
```

Si funciona, el servidor enviará el certificado al navegador par• establecer una conexión segura. Esto hará que el navegador le inste a aceptar el certificado ya firmado. Si se tratara de un certificado de una autoridad de certificación en la que el navegador ya conffa, no le preguntará nada .. En nuestro caso, creamos y firmamos nuestros propios certificados. No hemos optado por adquirir uno directamente, ya que primero queríamos comprobar que todo funcionara correctamente.

Si utiliza Internet Explorer o Mozilla, comprobará que aparece un símbolo en la barra de estádo que indica que se ha establecido una conexión segura. El icono utilizado en Netscape se muestra en la figura A.3.

Figura A.3. Los navegadores Web muestran un icono para Indicar que la página que estamos viendo proviene de una conexión SSL.

Últimos pasos

Para poder ver el objeto compartido PDFlib asf como otros módulos que haya instalado de esta forma, necesitará realizar algunos pasos adicionales. Copie el archivo libpdf_php al directorio de extensiones PHP, que probablemente será **/usr/local/lib/php/extensions**. Añada la siguiente línea a su archivo **php.ini**:

```
exteneion • libpdf_pbp.•o
```

Instalar Apache, **PHP y** MySQL en Windows

En Windows, el proceso de instalación es ligeramente distinto ya que PHP se configura bien como una secuencia de comandos CGI (php. exe) o como módulo

ISAPI (phpSisapi .dll). Sin embargo, Apache y MySQL se instalan de forma similar a la que hemos visto en Unix. Compruebe que ha instalado los últimos parches de servicios del sistema operativo en el equipo antes de comenzar la instalación de Windows.

En primef lugar debería descargar los últimos archivos fuente a un directorio temporal con espacio suficiente. En nuestra instalación hemos utilizado e : \ temp \ download como directorio temporal.

Si dispone de una conexión de red lenta, puede optar por utilizar las versiones del CD aunque es muy probable que se trate de versiones anteriores a la actual.

Instalar MySQL en Windows

Las siguientes instrucciones hacen referencia a Windows XP. En primer lugar configuramos MySQL. Suponiendo que ha descargado todos los archivos necesarios, descomprima el archivo zip MySQL en el directorio temporal y ejecute el programa Setup .exe. El instalador es un asistente lnstallShiedl convencional similar a muchos otros que haya utilizado.

Al seleccionar la instalación típica en el asistente, tendrá que indicar dónde quiere instalar MySQL El directorio de instalación predeterminado será e: \myaql, aunque puede cambiarlo a un directorio distinto una vez instalado, pero con algunos pasos adicionales para garantizar que todo funciona.

Si cambia de posición MySQL y pretende ejecutar el ejecutable myaqld, tendrá que indicarle por medio de opciones de linea de comandos dónde se encuentra todo. Utilice e: \mysql \bin \111ysqld --help para mostrar todas las opciones. Por ejemplo, si ha cambiado la distribución MySQL a 'D: \prograu \mysql' tendmqueiniciarmysqldcon 'D: \programs\myaql\bin\mysqld ~-basedir D: \programa\ mysql'. Si lo cambia y lo ejecuta como servido de Windows, tendrá que aear un archivo ini con el nombre my. ini y ubicarlo en su directorio de Windows principal. Su archivo ini tendrá un contenido similar a éste:

```
(myeqld)
ba•edir•D:/progr .../my•ql/bin/
datadir• D:/program•/myeql/data/
```

En la configuración de NT/2000 /XP, el nombre del servidor MySQL es mysqld-nt y normalmente se instala como servido. Un servicio es un programa que se ejecuta constantemente de fondo y cuya función es propordonar servicios para otros programas. Se ejecutan automáticamente al iniciar el equipo, lo que evita tener que iniciarlos manualmente cada vez. Puede instalar el HrVidor MySQL como servicio si escribe lo siguiente en la línea de comandos de Windows:

```
cd e: \-reql \bin
::iyeqld-nc -in•call
```

Obtendrá la sigui~~ respuesta

```
service •ucce••fully in•called.
```

EL_HOLISTICO@HOTMAIL.COM - 2006

Ya puede iniciar y detenerel servicio MySQL desde la lfnea de comandos con·

```
•rr START ll}'aql
DT  STOP ayaql
```

Ffjese en que el nombre del ejecutable;; ~--;qld·nt, pe~-~no~b~ del servido simplemente es myaql. Si ejecuta NET START m>:9Cllver4 el siguiente mensaje:

```
Tbe MySql aervica waa atarted aucceaafully.
```

Una vez instaladoel servido,se puede detener, iniciar o configurarpara que se inicie automiticamentepor medio de la opción Servicios del Panel de control. Para abrir Servicios, seleccione lnlcio>Configuración>Panelde control. Pulse dos veces sobre Herramientasadministrativas y dos veces sobre Servicios.

La ~tilidad Servicios se muestra en la figura A.4. Si quiere configuraralguna de las opaones de MySQL, primero debe detener el servido y, tras ello, especificarlas como par,metros de inicio antes de reiniciar el servicio MySQL Este servido se puede detener por medio de la utilidad Servicios o con los comandos NBT STOP MySQLo myaqladmin ahutdown.

Figura A.4. La utilidad S8fYIclos le permite configurar loe s8fYIclos que se ejecutan en su equipo.

Para probar si MySQL funciona o no, puede ejecutar el siguiente comando:

```
C:\ll}'aql\bin\ll}'aqlabow
```

La confi~ón pred~erminada no es la mú idónea. Hay algunos cabos sueltos que conviene resolver:

- Configurar la ruta
- Eliminar las cuentas anónimas
- Definir la contrasetla raíz

Configurar la ruta

MySQL incorporamultitud de utilidades de lfnea de comandos de distinta utilidad. No resulta nada sencillo obtenerlasa menos que el directorio binario de MySQL se encuentreen su PATH. Elobjetivode esta variablede entorno es indicar a Windows dónde debe buscar los programas ejecutables. Mucho de los comandos habituales que se utilizan en la línea de comandos de Windows, como por ejemplo dir y cd, son internos y se incorporan encm. exe. Otros, como fonat e ipconfig, disponen de sus propios ejecutables. Noserfaconveniente tenerque escribirC: \lfINNT\ ayetem32\format para aplicar formato a un disco ni tampoco e: \mysql \bin\ myeql para ejecutar el monitor MySQL El directorio en el que se almacenan los ejecutables de los comandos búicos de Windows, como format. exe, se encuentra automiticamenteen su PATH, por lo que basta con escribir format. Para disponer de la misma convenienciafunci6ncon las herramientas de lfnea de~dos mysql, será necesario añadirla. Pulse laido y selecdone Conflguración>Panel de control. Pulse dos veces sobre Sistema y seleccione la ficha Opciones avanzádas. Si pulsa el botón Vanableede entmno, se abrir6 un cuadro de di4logoen el q__ u• puede ver las variables de entorno de su sistema. Si pulsa dos veces sobre PATB podr, modificarla. Ai\ada un punto y coma al final de su ruta actual pare ~ la nueva entrada de la anterior y, tras ello, aAlda e : \myaql \bin. Al ~r Acleptu, se almacenan en el registro del sistema. Cuando reinicie el equipo; podrá escribir mysql en lugar de e: \myaql \bin\myaql.

Eliminilr el usuario anónimo

La configuraciónpredeterminadade MySQL permite que cualquier usuario acceda al sistema sin necesidad de proporcionar un nombre de usuario o una contraseila, lo que no es aconsejable. Lo primero que queremos hacer es borrar el usuario anónimo. Para ello basta con abrir una lfnea de comandos e introducirlas siguientes líneas:

```
c:\ll}'•ql\bin\ll}'aql
ua• ll}'aql
delete froa ua•r wure Ua•r•''1
quit
c:\myaql\bin\myaqladlllin reload
```

De esta forma se elimina el usuario anónimo.

Definir la contraseña raíz

La cuenta de superusuario, el usuario rafz, todavía no tiene contrasefta. Para configurarla, escriba estas líneas:

EL_HOLISTICO@HOTMAIL.COM - 2006

```
c:\ay•ql\bin\ay•qladmin -u rooc P•••vord your_p•••vorcl
c:\iay•ql\bin\ayeqladllin -u rooc -b your_boec_na.. pae•vord your_pa••vord
```

Comprobaráque ahora, las tareas que anteriormenteno requerían un nombre de usuario y una contraseña, fallan sin esta información. Si intenta ejecutar

```
c:\ayeql\bin\ayeqladain  reload
```

o

```
c:\ayaql\bin\ayaqladllin  •bucdown
```

no lo conseguirá.

A partir de ahora, tendrá que utilizar el indicador -u y proporcionar un nombre de usuario, y el indicador -p para indicar a MySQL que dispone de una contraseña, como en este ejemplo:

```
e:\ay•ql\bin\ayeqladmin -u rooc -p reload
```

Si utiliza este comando, MySQLsolicitará la contraseña raíz que acaba de definir. Si necesita más información, puede consultar el sitio Web de MySQL ht tp: *11-w.* myaql. coaa. Ya podemos instalar Apache en Windows. Manos a la obra.

Instalar Apache en Windows

Apache 1.3' y posteriores se han diseftado para su ejecución en Wiridows NT, 2000 y XP. El imtaiador sólo funciona con la familia de procesadores x86, como los de lntel. Apadle tambim se puede ejecutar en Windows 85, 98 y ME pero estos sistemas operativos no se han probado. En c:Ílalquier caso, es necesario instalar TCP/IP. Asegúrese de utilizar la biblioteca Winsock 2 si decide realizar ia instalación en Windows 95 o 98. Desplácese hasta ht tp: / /ht tpd. apache ..org y descargue el binario de Windows correspondientea la versión actual de Apache 1.3 (Apache 2.0 utiliza subprocesos y algunas bibliotecas externas de PHP no admiten los subprocesos; por lo que le recomendamos que utilice la versión 1.3).

En nuestro caso hemos descargado el archivo apache_1.3.31-winl 2-x86 - no_are . msi. Contiene la versión actual (dentro de ia jerarquía 1.3) para W'mdows, sin código fuente, empaquetado como archivo MSI. Los archivos MSI son el formato de paquete utilizado por el instalador de Windows. A menos que tenga un problema dificil de solucionar o quiera contribuir a los esfuerzos de progrUIUICión, no es aconsejable que compile personalmente el código fuente. En este archivo se incluye el servidor Apache listo para su instalación. Para iniciar la instalación basta con pulsar dos veces sobre el archivo descargado. El proceso de instalación le resultará familiar. Como puede apreciar en la figura A.5, es similar a muchas otros instaladores de Windows. El programa de instalación le solicitará la siguiente información:

- El nombre de la red, el nombre del servidor y ia dirección de correo el~nico del administrador. Si tiene pensado aear un servidor para su uso real, debería con~r las resl:'uestas a estas preguntas. Si lo quiere aear para uso personal, esta informaaón no es realmente importante.

- Indicar si quiere ejecutar Apache como servido. Al igual que con MySQL es más sencillo configurarlo de esta forma.
- El tipo de ~talación. Recomendamos ia opción Complete (Completa), pero puede elegir Custorn (Personalizada) si quiere ignorar determinados componentes, como por ejemplo ia dóc:wnentación.
- El directorio en el que desea instalar Apache (el predeterminado es C: \Archivos de progra- \Apache Group\Apache).

Una vez seleccionadas estas opciones, se instalará y se iniciará el servidor Apache. Apac~ se co~unica con el puerto 80 (a menos que haya cambiado las directivas ~ort, Listen o BmdAddress en los archivos de configuración). Para conectarse al servidor y acceder a la página predeterminada, inicie un navegador e introduzca este URL: http: //localho•t/.

Accederá a una página de bienvenida similar a la de la figura A.1, con un enlace al manual de Apache. Si no sucede nada o si obtiene un error, revise el archivo error. log del din!ctorio log•. Si su host no está conectado a Internet puede que tenga que utilizar este otro URL: http: / /127. o. o. 1. Es la direcci~ JP equivalente al equipo locaL Si ha cambiado el número de puerto, tendrá que adjuntar *:n~..JIIIffIO* al final del URL Debe saber que Apache no puede compartir el uusmo puerto con otra aplicac:i6nTCP/IP.

. Puede iniciar y detener el servido Apache desde el menú Inicio. Apache se adJunta a Programas>~r HTTP Apache. Debajo del título Control Apache Server encontrari opciones para iniciar, detener o reiniciar el servidor.

EL_HOLISTICO@HOTMAIL.COM - 2006

Una vez instalado, puede que sea necesario modificar los archivos de configuración del directorio conf. Veremos cómo se modifica el archivo de configuración bttpd.conf cuando instalemos PHP.

Si necesita activar Apache con SSL en Windows, consulte la sec:c:i6n de preguntu mú frecuentes de la direc:c:ión http: //tud.at/progra–/apacbe-aal-win3 2-bowto. pbp3, aunque no es nada sencilla.

Instalar PHP en Windows

Para instalar PHP en Windows, primero debe descargar los archivos de PHP 5 en la dirección http://www.php.net. Para la instalación en Windows debe descargar dos archivos. Uno es el archivo ZIP que contiene PHP (con un nombre como pbp-5. o. o-win32 .zip) y el otro es una colección de bibliotec:u (pecl-5. o. o-Nin32. zip o algo similar). En primer lugar, descomprima el archivo Zip en un directorio de su elección. La ubicación habitual suele ser e: \PHP, la que utilizaremos en nuestra descripción. Puede instalar lu bibliotec:u PECL si descomprime el archivo PECL en su directorio de extensiones que, si utiliza e: \PHP como directorio base, seria e: \ PHP\ext \. Tru ello, siga los pasos descritos a continuación:

1. En el directorio principal veri un archivo con el nombre pbp. exe y otro denominado pbp4ts. dll. Son los archivos que necesita para ejecutar PHP como CGI. Si quiere ejecutarlo como módulo SAPI, tendm que utilizar el correspondiente archivo DLL de su servidor Web. Si utiliza Apache, el archivo se denomina pbpSapacbe. dll. Los módulos SAPI son mú rápidos y mú sencillos de proteger; la versión CGI le permite ejecutar PHP desde la línea de comandos. La decisión es personal.
2. Copie todas lu DLL al directorio de sistema de Windows. En el caso de Windows NT o 2000 se encuentra en C: \winnt\system32 y en Windows XP, en e: \wi'ndows\system32.
3. Defina un archivo de c:onfiguración pbp. ini. PHP incorpora dos archivos preparados: pbp. ini-dist y php. ini-reco — nded. te sugerimos que utilice este último. Copie este archivo y cambie su nombre por pbp. ini. Guárdelo en el directorio ISISTBM-ROOTI. Normalmente se encuentra en c:\winnt o c:\winnuo en Windows NT o 2000, o en c:\windowa en WindowsXP.
4. Modifique su archivo pbp. ini. Incluye multitud de pammetros, la mayoría de los cuales puede ignorar. Los que debe cambiar son los siguientes:
   - Cambie la directiva extension_dir para que apunte al directorio en el que se encuentran sus DLL de extensión. En la instalación normal se corresponde a e: \PHP\ext. De esta forma, su archivo php. ini contendrá

     ```
     exteaaiaa_dir • c"/php/ext
     ```


   - Cambie la directiva doc_root para que apunte al directorio raíz desde el que adl1a su servidor Web. Probablemente será

     ```
     doc_root • •c,/Progrea rile•/Apecbe Gr:oup/Apecbe/btdoc•"
     ```

     si utiliza Apache, o

     ```
     doc_root • •c:/Inetpub/wwwroot•
     ```

     si utiliza ns.
   - También puede seleccionar lu extensiones que quiere ejecutar. Le sugerimos que, en esta fase, simplemente ponga PHP en funcionamiento. Podrá ai\adir las extensiones cuando lu necesite. Para añadir una extensión, consulte la lista incluida bajo "ixtensiones de Windows". Verá numerosas lfneu c:omo <ListadOS>;extension=php_pdf.dll

     Para activar esta extensión, buta con eliminar el punto y coma que aparece al principio de la linea (y realizar lo contrario para desactivarla). Si posteriormente desea añadir mú extensiones, tendrá que reiniciar su servidor Web después de cambiar pbp. ini para aplicar los cambios.

     En este libro utilizamospbp _pdf. dll, pbp_gd2. dll, pbJl!i••P. dll y pbp_mysqli . dl 1. Debería anular los comentarios de esta.aneas. Si falta php_mysqli .dll, ..Udala como se indica a continuación,,-

     ```
     return $this->$name;
     ```

     Cierre y guarde su archivo php. ini.
5. Si utiliza NTFS, asegúrese de que ei usuario con el que se e;eéfta el servidor Web dispone de los permisos necesarios para leer su archivo pbp. ini.

Añadir PHP a su configuración de Apache

Puede que sea necesario modificar uno de los archivos de c:onfiguraciónde Apache. Abra el archivo bttpd. conf en su editor preferido. Habitualmente este archivo se encuentra en el directorio e: \Archivos de Progra•a\Apache Group\Apacbe\conf\. Busque las siguientes lfneu:

```
LoaclNodul• phpS_aodule c:/pbp/pbpSepecbe.dll
AdclNodule aod_pbpS.c
AddType eppUcatl.on/x-bttpd-pbp .pbp
Action •pplicatiOD/X•bttpd-pbp•/pbp/pbp.exe•
```

Si no las encuentra, aMdalu al final del archivo, guárdelo y reinicie su servidor Apache.

Añadir PHP y MySQL a Microsoft 115 y PWS

En este apartado veremos cómo aAadir compatibilidad con PHP y MySQL a 115 c:on el módulo ISAPI(pbpS isapi •dll). Tambiél puede instalarlo como CGI, pero

EL_HOLISTICO@HOTMAIL.COM - 2006

le recomendamos que utilice el módulo ISAPI ya que es más r4pido. Se supone que habd realizado los pasos descritos anteriormente en los apartados sobre instalación manual o con lnstallShield. La principal diferencia es que la directiva de configuración doc_root sed e: /Inetpub/wwwroot. Tras ello, tendd que iniciar la Consola de administración de Microsoft (puede que en su sistema aparezca como Servicios de Internet lnformation Server). Si utiliza 2000 o XP, se encuentra en el Panel de control, bajo Herramientas administrativas.

Una vez abierto, en la parte izquierda ver4 una vista en 4rbol de los servicios. Pulse con el botón derecho del ratón sobre el servidor Web (normalmente con el nombre Servidor Web predeterminado) v seleccione Propiedades.

Se abrir4 el cuadro de dialogo Propiedades, en el que se incluye gran cantidad de información, pero de la que sólo cambiaremos parte. En la ficha Directorio particular, pulse el botón Configuración. En Asignaciones para la aplicación, pulse Agregar para añadir PHP. Le pedir4 un ejecutable, por lo que debe indicar la ruta completa de la ubicación de phpS isapi. dll (probablemente e: \php\sapi \phpS isapi . dl 1 si ha seguido las instrucciones). También ved un campo Extensión en el que debe introducir . php. Marque la casilla Motor de secuencias de comandos y pulse Aceptar.

Si quiere realizar autenticación HTTP (como hemos visto en el libro), debe utilizar la ficha Filtros ISAPI. Pulse Agregar. Se le pedir4 un nombre de filtro, escriba PHP, y un ejecutable, la ruta completa al archivophpS isapi. dll, como en el caso anterior. Pulse Aceptar. Cierre el cuadro de di4logo Propiedades por medio del botón Aplicar.

Detenga el servidor Web (o compruebe que est4 detenido) y reinícielo. Puede hacerlo desde la Consola de administración/Servicios de Internet lnformation Server si pulsa con el botón derecho del ratón sobre el servidor Web y selecciona la opción Detener. Puede reiniciarlo de la misma forma si pulsa lmdu.

Pruebas

Inicie el servidor Web y compruebe si PHP funciona. Cree un archivo test. php y añ4dale las siguientes líneas:

```
e? phpinfol)l ?>
```

Asegúrese de que el archivo se encuentra en el directorio raíz de documentos (normalmente en c:\Program File\Apache Group\Apac:he\htdocs bajo Apache oc: \Inetpub\wwwroot bajo DS) y 4bralo en el navegador como se indica a continuación:

```
http://localho•t/te•t.php
```

o

```
http,//~r-ip·llllllber-here/teat.php
```

Si accede a una p4gina similar a la de la figura A.2. significa que PHP funciona.

Instalar PEAR

PHP S incorpora el instalador del paquete PEAR (Repositorio de aplicaciones y extensiones de PHP). Si utiliza Windows, introduzca lo siguiente en la linea de comandos

```
c:\php\go-pear
```

La secuencia de comandos 90-pear le preguntar, dónde quiere instalar el paquete y las clases est4ndar de PEAR, y las descargad e instalar4 por nosotros.

Es necesario que cuente con una versión instalada del instalador del paquete PEAR. De esta forma, para instalar los paquetes, basta con que escriba:

```
pear inatall paquete
```

donde *paquete* se sustituye por el nombre del paquete que desee instalar.

Para ver la lista de los paquetes disponibles, escriba:

```
pear liat-all
```

Para ver los componentes instalados actualmente, pruebe con

```
pear list
```

Para instalar el paquete de correo MIMEque utilizamos en capítulos anteriores, escriba lo siguiente:

```
pear install Mail_Mime
```

El paquete DB mencionado en uno de ios primeros capftulos, se instala autom4ticamente, pero para comprobar que dispone de la versión m4s reciente, escriba:

```
pear list·upgrades
```

Si hay una nueva versión disponible, escriba

```
class classnamc
{
```

Si en su caso no funciona el procedimiento anterior, pruebe a descargar directamente los paquetes PEAR en lugar de hacerlo a travá del instalador. Para ello, puede visitar la p4gina http://pear.php.net/packages.-php.

Desde aquí, puede examinar los distintos paquetes disporubles. Por ejemplo, en este libro hemos utilizado Mail_Mime. Pulse en la p4gina de este paquete *y* pulse sobre Download Latest para obtener una copia. Tendr4 que descomprimir el archivo descargado y guardarlo en su ruta de incl~ión..

También ser4 necesario contar con un directOno c: \php\pear o smular. 5l descarga manualmente los paquetes, es aconsejable que los guarde en el úbol de directorios de pear: PEARdispone de una estructura est.indar por lo que le sugeri-

EL_HOLISTICO@HOTMAIL.COM - 2006

mos que almacene los componentes en la ubicación estmu:lar, es decir, donde los guardaría el instalador. Por ejemplo, el paquete Mail_Mime corresponde a la seeción Mail. por lo que en este ejemplo lo guardaríamos en el directorio e: \php\pear\Mail.

Configuraciones adicionales

Puede configurar PHP y MySQL con otros servidores Web a,mo Omni, HTPPD y Netscape Enterprise Server. No los describiremos en este apfndice pero en los sitios Web de MySQL y PHP puede encontrar información sobre cómo se configuran, http://www.php.net y http://www.mysql.com, respectivamente.

EL_HOLISTICO@HOTMAIL.COM - 2006

Apéndice B
Recursos ~JVeb

En este apéndice enumeraremos algunos de los muchos recursos disponibles en la Web, que puede utilizar para buscar cursos prácticos, artículos y ejemplos de código PHP. Se trata de una pequei\a selección. Evidentemente hay muchos más de los que podríamos incluir en un apéndice. Y otros muchos que aparecen diariamente debido al aumento del uso y la popularidad de PHP y MySQLentre los programadores Web. Algunos de estos recursos estarán en idiomas diferentes, como alemán o francés. Le recomendamos que utilice un traductor como ht tp: / /www. syntransoft. compara explorar el recurso Web en su idioma nativo.

Recursos PHP

- PHP.Net (ht tp: / /www. php. net). El sitio original de PHP, en el que puede descargar todas las fuentes de PHP así como una copia del manual.
- ZEDN.Com (http://www. zend. com). Fuente del motor ZEND que dirige PHP. Un portal que contiene foros, artículos, cursos prácticos y una base de datos de clases y código de muestra que puede utilizar. No se la pierda.
- PEAR(http: / /pear. php. net): El Repositorio de extensiones y aplicaciones de PHP. Es el sitio oficial de extensiones de PHP.
- PECL(ht tp: / /pecl. php. net): Sitio hermano de PEAR. PEARse ocupa de clases escritas en PHP; PECL se encarga de las extensiones escritas en C. Las

clases PECL suelen ser más difíciles de instalar pero ofrecen mejoft!Sfunciones y resultan más potentes que sus equivalentes de PHP.

- PHPCommunity (ht tp: / /www. phpcommunity. org): Un nuevo sitio basado.en una comunidad.
- php | architect (ht tp: / /www. phparch. com): Revista sobre PHP. Incluye artículos gratuitos aunque también se puede suscñbir para recibir los ejemplares en PDF o en formato impreso.
- PHP Magazine (ht tp: / /www.phpmag. net): Otra revista sobre PHP, también disponible en formato electrónico o impreso.
- PHPWizard.net (http://www.phpwizard.net).Fuente de numerosas y atractivas aplicaciones de PHP como phpChat y phpIRC.
- PHPMyAdmin.Net (http://www.phpmyadmin.net): Sitio de la conocida interfaz Web basada en PHP.
- PHPBuilder.com (http://www. phpbuilder. com).El portal de cursos prácticos sobre PHP. En este sitio encontrará cursos prácticos sobre todo lo que se le ocurra. También dispone de un foro y un tablón de anuncios en el que los usuarios pueden dejar mensajes.
- DevShed.com (ht tp: / / devshed. com). Portal que ofrece cursos prácticos sobre PHP, MySQL, Peri y otros lenguajes de desarrollo. Imprescindible para principiantes.
- PX-PHP Code Exchange (http: / /px. sklar. com). Un buen punto de partida. Encontrará muchos ejemplos de secuencias de comandos y también funciones de gran utilidad.
- The PHP Resource (ht tp: / /www. php-resource . de). Sitio con cursos prácticos, artículos y secuencias de comando. El "único" problema es que está en alemán. Le recomendamos que utilice un servicio de traducción para verlo.
- WeberDev.com (http://www.WeberDev.com). Anteriormente conocido como la página de ejemplos de PHP de Berber, este sitio ha crecido y en la actualidad incluye cursos prácticos y código de ejemplo. El sitio está dirigido a usuarios de PHP y MySQL, y también cubre aspectos relacionados con bases de datos de seguridad y generales.
- HotScripts.com (http://www.hotscripts.com). Excelente selección de secuencias de comandos por categorías. Dispone de secuencias de comandos para diferentes lenguajes, como por ejemplo PHP, ASP y Peri. También cuenta con una amplia colección de secuencias de comandos de PHP. Se actualiza frecuentemente. Visita obligada si lo que busca son secuencias de comandos.
- PHP Base Library (ht tp: / /phpl ib. sourceforge . net). Sitio utilizado por programadores de proyectos de PHP a gran escala. Cuenta con una biblioteca con multitud de herramientas para enfoques alternativos de administración de sesión, así como con plantillas y abstracción de bases de datos.

- PHPCenter (http://www.php-center.de). Otro portal alemán utilizado para c:msos pnlc:licossecuencias de comandos. comejos, trucos, publiddad, etc.
- PHP Homepage (http://www.php-homepage.de). Otro sitio alemán sobre PHP con secuencias de comandos, utfculos, noticias y mucho más. Dispone de una sección de referencia.
- PHPindex.c:om(ht tp: / /www. phpindex. com). Portal francés sobre PJfP con gran cantidad de contenidos relacionados con PHP. Cuenta con noticias; lí-bros, sección de preguntas más frecuentes, artfculos, etc.
- WebMonkey.com (ht tp: / /webmonkey. cOII). Portal con multitud de recursos Web, cursos pnlcticos reales, ejemplos de código, etc. El sitio cubre aspectos relacionados con el diseño, la programac:i6n, sistemas, multimedia y muchos otros.
- PHP Club (http://www.phpclub.net). O&ece una gran cantidad de recursós para principiantes de PHP. El sitio dispone de noticias, críticas de libros, ejemplos de código, foros, sección de preguntas más frecuentes y multitud de cursos prácticos.
- Repositorio de clases PHP (ht tp: / /www. phpcluses upperdesign. com). Sitio que distribuye clases gratuitas escritas en PHP. Es una visita obligatoria si tiene pensado desarrollar su código o si su proyecto está formado por clases. Dispone de funciones de búsqueda, para localizar con facilidad lo que necesite.
- PHP Resource Index (ht tp: / /php. resourceindex. com). Portal que ofrece secuencias de comandos, clases y documentación. Lo más atractivo de este sitio es que toda la información se agrupa en categorías, lo que nos permite ahorrar mucho tiempo.
- PHP Developer (http://www.phpdeveloper.org). Otro portal de PHP que ofrece noticias, artículos y cursos prácticos sobre PHP.
- Evil Walrus (ht tp: / / evil wal rus com).Atractivo portal en el que encontrará secuencias de comandos de PHP.
- Source Forge (http: //sourceforge.net). Gran cantidad de recursos de código abierto. No sólo le permite buscar código que le resulte útil, sino que también ofrece acceso a CVS, listas de correo y equipos para programadores de código abierto.
- Codewalkers (ht tp: / / codewalkers. com). Contiene artículos, criticas de libros, cursos prácticos y el sorprendente Concurso PHP, en el que puede ganar por medio de sus nuevos conocimientos. Ofrece un concurso nuevo cada dos semanas.
- PHP Developer's Network Unified Forums (http: / / forums. cievnetwork . net/index.php). Análisis de todos los aspectos relacionados con PHP.

EL_HOLISTICO@HOTMAIL.COM - 2006

- PHP Kitchen (http: / /phpki tchen. c01D). Artículos y noticias sobre PHP.
- Postnuke (http://www. postnuke. c01D); Sistema de gestión de contenidos de PHP muy utilizado.
- PHP Application Tools (http://www.php-tools.de). Conjunto de clases de PHP muy útiles.

Recursos específicos de MySQL y SQL

El sitio de MySQL (http://www. mysql. c01D). Sitio oficial de MySQL Ofrece excelente documentadón, asistencia e información. Visita obligatoria si utiliza MySQL. SQL Course (http://sqlcourse.com). Cuenta con ůn curso práctico sobre SQL con instrucciones muy sencillas de entender. Le permite practicar lo que ha aprendido en un intérprete SQL en linea. Puede encontrar una-versión avanzada en http://www. aqlcourse2. com.

SearchDatabue.com (http://searchdatabase.com). Atractivo portal con multitud de información sobre bases de datos. Cuenta con excelentes c:unos prácticos, consejos, sección de preguntas mú frecuentes, críticas, etc. Imprescindible.

$tn_height = $max_height;

Apache Softthre (http://www. apache. org). Punto de partidasi necesita descargar el código fuente o l0S aich1vos.bínaños-EI sitio también cuenta con documentadón en linea.

Apache Week (http://www. apacheweek. com). Revista en linea semanal que ofrece infonnac:ión fundamental para todo el que utilice un servidor Apache o para cualquiera que ejecute servicios Apache.

Apache Today (http://www. apachetoday. com). Noticias e informad6n diaria ~re.~P~ Los usuarios deben suscribirse para poder enviar preguntas.

oeta:~rollo web

Philipand Alex'sGuideto Web Publishing (http://philip.greenapun.com/panda). Una inteligente e irreverente gufa sobre ingenierfa de software aplicada a la Web. Uno de los pocos libros sobre el tema eecrito por un Sunoyedo.

EL_HOLISTICO@HOTMAIL.COM - 2006

El CD-ROM que acompaila al libro contiene versiones completas de P.flP, MySQL, Apache, varias bibliotecas gr4ficas y archivos que contienen los listac:loide código del libro, distribuidos de la siguiente manera:

Ejemplos

En este directorio encontrar,, organizados por carpetas por cada uno de los capítulos, los listados de ejemplo que se han desarrollado a lo largo del libro.

Software

Contiene las siguientes herramientas:

- IMAP: Internet Message Access Protocol (IMAP) es un método de acceso al correo contenido en un servidor de correo.
- PDPLib:API para generar archivos PDF desde el lado del servid~ o del cliente.
- mocl..audunysql: módulo de autenticación de Servidor Apache, haciendo uso de una base de datos MySQL para el almacenamiento de nombres de usuarios y contrasellas para un acceso mú rápido.
- GnuPG: reemplaza, de forma completa y gratuita, a PCP.
- gd: librerfa gr4fica que contiene libpng, zlib y h"bjpeg.

EL_HOLISTICO@HOTMAIL.COM - 2006

- Adobe Aaobat Rader: populu visor de documentos PDF.

Adeims, encontrad las versiones completas de Apache, MySQL Y PHP en sus respectivu carpeta. Tenga en cuenta las siguientes consideraciones:

Windows

En el Apéndice A se describe la configuración de Apache, MySQL *y* PHP en una plataforma de Wlndows.

Apache 1.3.31 se encuentra en el directorio Software\Apache\Window•\Binary(apache_1.3.31-win32-x86-no_src.exe).

Tanto la venión de producción actual de MySQL (4.0, ·mysql-4. o. 20c-win. zip) *y* la venión alfa (S.0, myaql-5. 0. Oa. alpha-win. zip) se encuentran en el directorio Soft-re \MySQL\Ilindow•\Binary. Descomprimalu *y* haga doble clic solne SBTUP ·BXB para iniciar el programa de instalación. Tras ello siga lu instruc:ciones del ApcfndiceA para preparar su instalación de MySQL*y* poder realizar los ejemplos del libro.

PHP *S* se encuentra en el directorio Software\PHP\Windows\Binary. Siga Ju instrucciones recogidas en el Apcfndice A para configurar PHP para su sistema.

!.inuxiUnhr

Muchas distribuciones de Linux *y* algunas estaciones de trabajo de Unix están configuradas con Apache, MySQL*y* PHP. Sin embargo puede que no sean la últimas venianes desuitd er, este libro. Enel Apéndice A se de9cribe la configuraci6n de Apache, MySQL*y* PH.P en una estación de trabajo Unix o Unux por si tiepe que instaladm. En el CD-ROM se incluye el código fuente para Apache, MySQL*y* PHP asf como instaladoreSbinarios para MySQL en Linux.

El código fuente para Apache 1.3.31 se encuentra en Software/ Apache/~ix/source. Si dispone de tar GNU, utilice ht tpd-1 . 3 ·31 . tar. gz. En caso contrario, utilice ht tpd-1. 3 . 31. tar. Z.

Los instaladores binarios para MySQL Max 4.0 *y* S.0 para Linux se encuentran en Software\MySOL\ODix\Binary. Si su sistema utiliza el administrador RPM para instalar software, utiliceMySQL-Max-4. 0 .20-0. i386 .rpmoMySQL-Max-5. 0 .c-0 . i3 86 • rpm para instalar la parte del servidor de MySQL,*y* utiliceMySQL-client-4 .O. 20-0. i386. rpmoMySQLclient-5. O. O·O. i386 .rpmpara instalar la parte del cliente. Si su sistema Unux no utiliza el administrador RPM para instalar software, utilicemyaql-max-4. 0 .20-pc-linuxi686. tar .gz omysql-standard-5. 0. O-tllpa-pc-llmx-i.686. tar .gzpmainslalar laspmesdec:lienleylelYidordeMySQL

El c:6digo fuente para MySQL 4.0.20 para Unix se encuentra en myaql-4. 0. :zo. tar. gz, *y* para la versión 5.0, en myaql-5 • 0.0-alpha.tar.gz. Los usuarias de Solaris deben descargar tar GNU para extraer estos archivos debido a un error en la veni6n de Solaris del programa tar.

Elc6digo fuemepara PHP S.0 se encuentra en Software/PHP/tJnix/Source.

Camulteel archivo u.me del CD-ROM para ampliar esta infonnaci6n.

EL_HOLISTICO@HOTMAIL.COM - 2006

ln r 0

A

Acceder
- a base de datos con PHP, 308
- a elementos de una matriz, 117
- a los contenidos de matrices, 115
- a variables de formulario, 44

Acceso anónimo, 425
ACI0,358
Actualizar
- registros, 285

Administrar
- la lecha *y* la hora, 494
- listas de correo, 32
- MySQL,246
- senicios de Internet. 424
- sistema,44

Agilizar consultas con indices, 346
Agregar contenido dinúnico, 44
Agrupar *y* agregar datos, 298
Ajustar el texto en el botón, 522
Algoritmo
- de clave secreta, 400
- de compresión, "40
- de comprobación aleatoria, 416
- de encriptaci6ii, 393
- hash, 440
- Message Digest *S*, 416
- Unix Crypt, 416

Allaire ColdFusion, 34
Almacenar
- contrasei\as, 413
- datos en cachl, 908
- datos restringidos,436
- en caché, 875
- marcadores, *6111*
- *y* recuperar datos, 88

Alterar tabla tras su aeación, 304
AMANDA,406
Ámbito,44
Ámbito de variables, 44
Amenazas contra la seguridad, *'3ffl*
Ampliarelproyecto,683
Analizar XML.875
Anclajes, 161
Añadir
- ampliadones, 836
- artfculos al carro, *66S*

contenido en el sistema, 689
datos a la bue de datos, 285
marcadores, 631
nueva informaci6n a la base de datos, 309
nuevos artfculos, 830
PHP
a su configuración de Apache, 937
y MySQLa Microsoft ns yPWS, 937
valor a los artfculos y servidos, 380
Apache, 34, 178
Aplicación
bookmark_fns.php, 609
Book-O-Rama, 250
Carro de la compra, 648
d,ta..valid_fns.php, 609
de administración de contenido, 692
de autenticación, 548
de BooJt-0-Rama, 259
de correo, 451
de correo electr6nico cifrado, 451
de diagrama de barru, 535
de evaluación real, 848
de formato, 255
de foros Web, 807
de gestión de listas de correo, 761
de Google, 437
de la operación UPDATE, 303
de listu de correo, 806
do_html_header, 727
include_fns.php, 723
index.php, 888
PHPBookmark, 609
principal, 888
principal about.php, 886
Pyramid-MLM, 765
require_once, 620
SmartForm, 155
Tahuáyo, 875
TCP/IP, 935
Warm Mail, 721
Web, 263
Aplicar
formato
a cadenas, 144
al resultado, 691
ingeniería de software al desarrollo Web, 567
una función a cada elemento de una matriz:, 134
Aprendizaje de PHP, 36
Archivo
.htaccess, 179
abrir, 89
abrir a través de FTP o HTIP, ~5
adjuntos HTML, 476
binario, 403
cerrar, 89
crear, eliminar y desplazar, 468
de claves, 442
de contenido, 455
de datos, 434
de listas, 785
dePHP, 434
descriptivos, 464
escritura, 588
httpd.conf, 316
lectura, 588
php.ini, 53
temporales, 450
Arquitectura, 246
de bue de datos Web, 254
de la secuencia de comandos, 725
ASCil, 277
Asipr valores a variables, 44
ASP, 34, 43
Atributo -
de acción, 452
de la clase actual, 203
de la subclase, 209
de su superclase, 209
dentro de una clase, 203
desde el exterior de una clase, 203
UNSIGNED, 278
ZEROFILL, 278
Auditorias y registros, 387
autenticación, 387
búica, 409
de texto implícita, 418
HTIP, 419
Autenticación, 604
básica con IIS, 424

Autenlicaa'ó,| y personaliz.ar usuarios, 32
Ayuda de depuraci6n de variables; 594

B

Bajo coste; 35, 37
Banco de pruebas, 44
Base de datos
frente a almacenamiento en archivos, 690
relacional, 32, 245
Biblioteca
incorporada, 35
PECL, 37
Bifurcación, 161
BillCates Wealth Cock. 479
Bloqueo
de archivos, 108
Bloques de código, 78
Bucle
for y foreach, 84
while, 83
Búsquedu, 710
cadenas en cadena, 155
coincidencia de c:aracteres especiales, 162
filu que no coincidan, 29'
subcadenu
con expresiones regulares, 164

e

CA, 402
Calcular, 440
fechas
en MySQL, 503
enPHP, 502
Calificar lu respuesta, 849
Cambiar
mayl1sculu y minl1sculu en una cadena, 147
propiedades de archi'l,:o, 468
Cambios
en linea, 689
Capa
de aplicación, 438
de host a red, 438
dered, 438
de transporte, 438
segura de sockets, 436
$L, 438
Caracteres especiales, 162
Caracteres inofensivos, 442
Carga
de archivos, 455
de matrices desde archivos, 130
de un nuevoboletfn informativo, 792
din6mica de extensiones, 560
Carga de archivos, 456
Carro de la compra, 32, 647
Cerrar
conexión, 486
sesi6n, 608
como usuario rafz, 270
Certificado digital, 387
Certifying Authorities, 402
Ofrar
contraseilas, 416
MySQL.'29
la informaci6n, 436
Case
dir, 203
Page, 213
Reflec:tion, 229
ServicesPage, 219
Clues
y obj¡etos, 198
Clave, 247
de sesión, 436
privada, 400
pllblica, 387
secundaria, 355
Clonar objetos, 198
Código
abierto, 31, 33, 35, 37, 38, 444
de autenticaci6n de mensaje, 440
de mensaje de autenticación, 440
fuente, 36, 38
HTML, 33, 45
PHP de carga de archivo, 457
Colocar el texto, 525
Columnas, 247

EL_HOLISTICO@HOTMAIL.COM - 2006

Comando
mysql,261
mysql_dump, 348
Combinación
de dos tablas, 292
de operadores de asignación, 65
de varias tablas, 294
y división de cadenas, 150
Comentar el código, 572
Comentarios, 48
Comercio electrónico, 31, 32, 34, 39, 43
seguridad, 386
y seguridad, 39, 369
Comparar
cadenas, 141
condiciones diferentes, 82
funciones
de cadenas, 165
de expresiones regulares, 165
Compatibilida4 36, 38, 377
con el enfoque orientado
a objetos, 36
SS(,428
Competencia excesiva, 381
Componentes de la solución, 666
Comprimir, 440
Comprobar-
existencia de un archivo, 106
incorrectamente los datos de
entrada, 592
longitud de una cadena, 154
tiempo de actualización
de archivos, 487
tipo de clase y sugerir tipos, 223
y filtrar datos entrantes, 313
Concatenar cadenas, 56
Conceptos
de base de datos relacionales, 246
deMySQL.33
dePHP,33
deSQL.285
delcontroldeexcepciones,231
del control de sesiones, 537
orientados a objetos, 197
Conclusión. 39, 640
y posibles ampliaciones, 640
Conectarse
a servicios Web con XML y SOAP,
874
al servidor FTP remoto, 487
Conexión
a redes de servicios, 591
SSL,396
Confianza,377
Configuración
adicional, 940
base de datos, 723
compatibilidad ele imágenes
enPHP,510
de cookies desde HTTP, 539
de cuentas, 734
deruta,933
de una conexión, 315
de usuario para la Web, 269
del control de sesiones, 545
del servidor o servidores
secundarios, 351
del servidor principal, 350
Conjuntos y clases de caracteres, 159
Constantes
ASSOCIATED, 891
CACHE,891
de clase, 222
de informe de error, 598
de tipo de error definidas, 599
definida,897
definidas, 61
DEVTAG,891
global METHOD, 901
globales, 886
clave, 891
METHOD. La, 891
predefinidas
combinadas, 597
dePHP,61
proplas, 61
SID,540
Constructores, 201
Consultas
de base de datos, 313
de una base de datos
desde la Web, 313
Contar elementos de una matriz, 135
Contraseftas, 336
Control
de acceso, 335
de sesiones, 418
detipos,59
Controlador
ODBC,35
Controlar
acceso con private y public, 205
excepciones, 230
visibilidad por medio de prívate
y protected, 207
Conversión
clases en cadenas, 228
de matrices en variables
escalares, 136
de tipos, 60
entre formatos de fecha de PHP
yMySQL,495
Cookies, 35, 435
Copia
de seguridad, 349
de archivos geaerales..406
de datos, 388
y restauración de bases de datos
MySQL;406
Correo
basura,374
electrónico
basado en la Web, 32
encriptado, 448
Corta fuegos, 388
Coste, 36, 170
Crear
aplicación Bob's Auto Parts, 44
aplicación Smart Form Mail, 141
base de datos, 258
base de datos Web, 258
bases de datos y usuarios, 262
carro de la compra, 642
clases, atributos y operaciones en
PHP,200
formulario de pedidos, 45
foros Web, 808
gestor de listas de correo, 756
im6genes, 509
índices, 259
instancias de clases, 202
lienzo,512
política de seguridad, 3ff1
proyectos PHP y MySQL
prácticos, 565
sitios Web interactivos, 31
tablas de base de datos, 259
un sencillo ejemplo de sesión, 542
un servicio de correó electrónico
basado en la Web, 718
un sistema
de administración
de contenidos, 686
de autenticación propio, 429
un sitio Web
comercial, 32
de comercio electrQnico, 370
una cuenta nueva, 736
una lista nueva, 790
variables declaradas
por el usuario, 44
y .&iminHdifft:terios, 464-
Crypt_DES, 429
CSP,689
CSR,403
Curso acelerado de PHP, 42
Cursores y estructuras de control, 364

D

Data Encryptation Standard, 400
Datos
comprimidos, 440
de MySQL. 434
Dbm,35
Decidir un objetivo, 371
Declarar
procedimiento almacenado, 361
y utilizar constantes, 61
Definir
base de datoe, 762
de listas y suscriptores, 758
convenciones de nomenclatura, 571
cookie,538
la contrasefta raíz, 933

EL_HOLISTICO@HOTMAIL.COM - 2006

transacciones, 357
usuarios y privilegios, 259
Depuración, 584
DES, 400
Desarrollo Web, 31, 33, 35, 37, 39, 45
Desbordamientos de bdfer, 394
Descargar el archivo, 489
Desconectarse de una base de datos, 319
Desencadenar errores propios, 585
Designar funciones, 183
Desplazarse dentro de un archivo, 107
Desplazarse dentro de una matriz, 133
Desplegar y r«Q>legar, 818
Destructores, 202
Detección de intrusos, 396
Determinar
lienzo base, 522
si almacenar o no nwneros de tarjeta de cúdito, 433
Devolver
valores de asignación, 64
alores desde funciones, 190
Dibujar o imprimir texto en la imagen, 514
Dirección
IP, 338
exclusiva, 410
Directiva
Auth_MySQL_Encryption_Types, 429
AuthType Basic, 423
de configuración
doc_root, 938
magic_quotes_gpc, 150
register_long_arrays, 54
de servidor, 424
mable_dl, 561
cr"tension_dir, 561
magic_quotes...gpc, 441
ma:11_ellecution_time, 561
memorv _li,mit, -162
php_válue, 6-19
magic_quotes_gpc, 649
Port, Usten o BindAddress, 935
session.use_transsid. >&o
uploacLtmp_dir, 462
-with-imap, 720
Diseñar
clases, 198
la base de datos, 693
una base de datos Web, 244
una interfaz de administración, 645
y presentar la solución, 692
Disponibilidad de asistencia Ncnica, 35, 36, 38
Dividir
cadenas con expresiones regulares, 165
el código, 574
Documentar
y compartir funciones internas, 575
Documentar nuestros proyectos, 577
Documento
de Microsoft Word, 841
con un objeto COM, 842
de procesador de texto, 841
de sólo lectura, 856
en memoria, 868
en texto ASCil, 841
generados por PHP, 156
Helio World, 863
HTML, 862
PDP, 837
portable, 843
PostScript, 843
principal, 690
RPC822, 143
rft, 853
RTP, 839
Web, 172
WSDL, 881
XML, 49

B

Ejecutar PHP como intérpreteCGI, 919
Ejemplo
Wsico, 361
de uso de GRANTy REVOICE, 268
El equipo del usuario, 434
El operador de tipo, 71

El operador iguales, 67
El problema. 93
Eliminar
COm!O, 748
el usuario an6nimo, 260
marcadonis;-634
registros de la base de datos, 285
tablas, 285
un ardlivo, 106
una base de datos entera, 307
una cuenta, 737
Empaquetar, 440
Encabezado TCP, 440
Enaiptaci6n. 440
de clave
privada, 400
pdblica, 400
Entorno de transmisión seguro, 438
Enumerar
categorfas, 653
los libros de una categoría, 656
Enviar
correo, 719
con archivos adjuntos, 759
el mensaje, 801
nuevo mensaje, 749
y rec:ilrir COm!OS electrónicos, 476
Equifax Sec:ure, 402
Equilibrio entre usabilidad, rendimiento, coste y seguridad, 387
.Errcns
de ejecución, 585
de programación, 585
de sincronización, 394
de software, 381
lógicos, 585
sinblcticos, 585
Esdneres de virus, 396
Escoger las filas que recuperar, 300
Escribir
código
mantenible, 567
para nuestra clase, 213
en un archivo, 97
texto en el botón, 526
Espacios en blanco, 48
Esquemas, 248
Estúldar
ANSI, 38, 281
de codificación, 570
Estilos de etiquetas PHP, 48
Estructura
búica de una función, 182
de documentos, 690
de una clase, 201
Etiqueta
de ancla, 800
de apertura de PHP, 218
de apertura del formulario, 452
de cierre, 51
de cierre de PHP, 183
de desarrollador, 891
de imagen, 518
de imagen HTML, 520
dePHP, 51
de programador, 891
PHP, 517
SRC, 518
XHlML, 144
Evaluar
cadenas, 556
expresiones, 44
formatos de documento, 840
Evitar
almacenamiento de datos redundantes, 251
di1eft01s-convarios atributos vados, 253
herencia y reemplazos con final, 210
tiempos de espera, -190
Examinar
la base de datos, 275
tipos de variables. 44
Excepciones
definidas por el usuario, 231
en ejemplo Bob's Auto Parts, 231
y otros mecanismos de control de errores, 240
EXPLAIN, 339
funcionamiento de las consultas, 340
Exposición de datos confidenciales, 389

EL_HOLISTICO@HOTMAIL.COM - 2006

Expresiones regulares, 158
Extensiones de nombre de archivo y
require(), 172

p

Facilidad de uso, 37,377
Fallos
de alimentación,comunicadón,
redes y distribución, 383
de hardware infonmtico, 381
Fiabilidad, 170
Ficha Errores penonalizados, 426
Filas,247
FilePro,35
Filtrar las entradas de los usuarios, 441
Finalizar la e;ecuadn, 557
Firmu digitales, 387
Formato
de documento portable,839
de imagen, 511
de los archivo, 99
de'procesadores de texto, 841
de texto enriquecido,839
Poros Wl!IJ~3%" -
Fracaso en la atracción de suficiente
negocio,381
Fragmentos de código
de httpd.conf, 928
FreeBSD,36
FTP,95
FTP/SCP, 689
Función
__construct(), 201
_get(), 204
_set(), 204
__toStringO, 228
basename(), 466
checkdate(), 500
chgrp(),468
chmod(), 468
chown(), 468
dale(),51
DATE_PORMAT(),501
datediff(),5CM
de agregación,299
de archivo, 488
de base de datos, 592
de bibliotecas de imagen, 506
de buscar y reemplazar cadenu, 156
de carga de archivos, 268
de comparación, 126
de control de error, 600
de edición de texto, 689
dimame(), 466
dl(),560
eval(), 556
exec(), 469
fdose(), 99
file_exists(), 106
filemtime(), 488
ñlesize(), 106
fopen(), 91
get_current_user(),560
get_extension_funcsO,560
geUoaded_extensions0,559
get_magic_quotes_gpc(), 150
get_magic_quotes_runtime(),556
getlutmod(), 560
Header(),516
highlight_string(), 562
ImageChar(), 516
ImageFilledRec:tangle()532
ImageString(),515
Image11'FBBox0, 524
hnageTI"FText(t523
mail(), 143
mkdir(), 464
mktime(), 498
msyqli_error(), 590
mysqli_connect(),316
now(),SCM
parse_url(), 482
set_magic_quotes_runtime(), 556
time(),498
touch(), 468
UNIX_TIMESTAMP,502
Funcionalidad Wsica de sesiones, 538
Funcionamiento de las arquitecturas de
bue de datos Web, 309
Funciones
de archivo, 106
de archivo útiles, 106
de cadena, 141
de calendario, 505
de control de sesiones propiu, 538
de directorio,455
de estado, 488
de archivo, 468
de fecha, 492
deFTP,478
de hash unidirecc:ionales, 416
de imagen,534
de PostScript,516
de red de PHP, 480
de TrueType, 516
FTP,484
GIF,512
PNG,512
y uso de mayúsc:ulas
y minúsculas,181
Funciones avanzadas orientada
a objetos de PHPS, 221
Fundamentos
de la encriptación, 398

G

Generación de documenlOSPDF, 33,
838
Generar
grffico ñnal, 513
imágenes,508
un cadlogo en lfnea, 644
un c:ertific:ado
con PDJllib,864
PDF a partir de una plantilla, 856
RTF,852
un documento PDF por medio
de PDFlib, 859
GIF, 510
GMT,496
GNU,33,250
GPG,444
GPL,34,883
Guardar datos para su lectura
posterior,90
Guardar el carro actualizado,667

H

Hash, 228
criptogr6fico unidireccional, 416
Herencia, 197
múltiple, 211
Herramientas
administrativu,424
Htdocs,442
HI"ML, 32, 44
atributo
action,45
align, 45
alt, 174
archive, 767
background, 174
bgcolor,45
border,4
cellpadding, 83
cellspacing, 174
char, 105
checlced, 345
dass, 71
clear,262
code,235
color, 78
cols,449
colspan, 46
content, 174
data, 184
datetime,506
dedare,87
dir, 70
enctype, 456
for,44
headers, 143
height, 128
href, 175
id, 166
Ianguage. 49
link, 174
maxlength, 45
mectia,299
method,45
mulliple, 865
name,45

EL_HOLISTICO@HOTMAIL.COM - 2006

object, 222
onchange, 740
rows,275
scheme,482
selec~414
size,45
span, 174
SIC, 128
start, 351
style, 174
summary,374
target,658
text, 45
title, 46
type,45
valign, 613
value,46
version, 262
width,45
paralacargadearchivos,456
HTIP,53
HTIP 1.1,418
Hyperwave,35

1

IBM,400
IDEA,400
Identificador,44
de color,514
de imagen, 514
de la conexión Fl'I', 488
de programador,891
de resultados MySQL,655
Identificadores
de MySQL,276
descriptivos,277
Identificar
al propietariode la secuenciade comandos,560
visitantes,409
y personalizarusuarios,606
115,34,398
1155,424
ImageMagic:k,509
Imagen
GIF,35,509
JPEG,510
PNG,510
IMAP,438
Implementar
_ almacenamiento y recuperad6n de marcadores,631
autenticación
con el control de sesiones, 546
de usuarios,614
autenticacióncon PHP y MySQL, 408
base de datos, 610
carro de la compra, 659
cattlogo en lfnea, 652
control de acceso, 409
control de versiones,568
estructuras de control,77
funciones
administrativas,790
de usuario, 778
herencia en PHP, 206
inicio de sesión,772
interfaces,211
iteradorese iteración, 226
métodos estttic:os, 222
pago,673
recomendaciones,636
recunión, 193
replicación,329
sesiones simples, 540
sistema,695
sitio búico, 611
transacciones seguras con PHP y MySQL,432
un sistema de pago,645
una interfaz de administración,676
Importancia de la informadón,387
Imprimir un resumenen la barra de encabezado, 667
Incremento y decrementoprevio y posterior,65
Incrustar
código PHP, Sa
PHP en HTML,44
Inetpub, 178
Informix, 35
Inicializar
matrices indexadas numéricamente, 115
una matriz, 117
Innobase,351
InnoOBHot Backup,351
Instalación
binaria, 920
desde código fuente, 921
Instalar
Apache en Windows,934
Apache,PHP y MySQLen Unix. 920
Apache, PHP y MySQLen Windows, 930
el código del proyecto, 914
GPG,445
MySQL,922
en Windows, 931
POFlib,924
PEAR,919
PHP,924
PHP en Windows,936
PHP y MySQL,918
Instalar mod_auth_mysql. *4TI*
Instrucción
OOL,360
de asignación, 65
de cadena, 555
de procesamientoSGML,49
Instrucciones
linc:lude,171
debucle,"83
dePHP,49
Integración de base de datos, 35
Interacción
con el entorno,472
con el sistema de archivos,454
y el servidor, 454
con MySQL u otras bases de datos, 588
InterBase, 35
Interfaz de usuario, 695
Internet, 32, 45
Explorer, 379
InformationServer5, 424
Invocar funciones,52
IP,338
IRIX,36
ISAM,356
Iteraciones,44

J

Java Server Pages, 34
JPEG,510
JSP,34

L

Lectura
adicional, 111
Leer
correo,719
desde
directorios, 462
un archivo, 101
linea a lmea, 103
todo el archivo, 104
un car6cter,105
una longitud arbitratja de bytes, 106
Liberar recursos,517
Licencia
comercial. 34, 37, 44t
de código abierto,3't_37
Limitaciones
de la capacidad del sistema, 382
Limpiar cadenas, 144
Linux, 34, 36, 250
Linux/Unix, 950
Llamadas
a funciones, 179
no definidas, 180
que no existen, 588
por referencia frente a llamadas por valor, 169

M

MAC,440
Mala presentación,372
Malas especificacionestdcnicas, 393
Manipulaciones de matrices,133
Manipular
cadenas y expresiones regulues, 140
im6genes, 699

EL_HOLISTICO@HOTMAIL.COM - 2006

Marcu de tiempo de Unix, 497
Matrices
con diferentes indices, 117
indexadu numéricamente, 113
multidimensionales, 113
MD5,401
Medios publicitarios en linea, 372
Mensaje de error, 70
Método
.htaccess, 179
_call(), 225
_clone(), 224
_set(), 219
__toString(), 229
abstractos, 36, 224
ASINSearch(), 897
browseNodeSearc:h(), 901
de almacenamiento lnnoDB, 38
de autentic:ac:ión, 390
de carga de archivos, 689
de clase, 197
est4ticos, 197
Mic:rósoft
Active Server Pages, 34
ln&emet laioanatioA S.,i,.,34. 919
SQL Server, 17, 286
Windows, 34, 36, 38
Windows XP, 34
Modific:adores
de ac:c:eso, 197
private y protected, 2'17
Modific:ar
c:onfigurac:ión de una cuenta, 7tU
contenido, 689
contraseilas,624
datos, 389
panmetros de informes
de error, 597
temporalmente el entorno
de ejecución, SSS
una cuenta existente, 737
Mostru
artfculos, 821
libros de una c:ategorfa, 894
marcadores, 633
Motor
de MySQL, 434
dePHP,255
Zend,33,37
Motores de almac:enamiento, 355
MSQL.35
MySQL,31,32,33,34,35,37,38,39,45
desde el punto de vista del sistema
operativo, 336

N

Navegador
del usuario, 434
Web,48
Negación de servido, 392
Netsc:ape, 391
Niveles de informes de errores, 596
No actualizar el sitio, 373
No realizar el seguimiento del éxito
del sitio, 374
No responder a la información
generada
por el sitio Web, 373
No suministrar infonnaci6n
importante, 372
Nota sobre los tipos de tablu, 276
Novedades
de MySQL 5.0, 38
de PHP 5.0, 37

O

Obtener
información
c:on SHOW, 339
sobre arc:hi~os, 465
sobre columnu c:on
DESCRIBE. 340
sobre el direc:torio actual, 464
sobre el entomo de PHP, 559
la fec:ha y la horaen PHP, 495
mú información sobre bues
de datos, 338
un c:lue AmazonResultSet,
896
ODBC, 35, 324
Opción 5ec:ure Hash Algorithm 1, 416
Open Database Connec:tivity
Standard, 35
Operaciones de clue, 206
Operaciones de registro
de usuarios, 396
Operador
:,222
–,66
?,44
O,70
+,120
=,70
=e, 153
=s=, 156
=>,70
AND,68
c:ast, 904
c:oma, 69
comodín, 289
de asignación, SS
de cadena, 64
de c:omparac:ión, 67
de conc:atenac:ión, S6
de ejecución, 70
de identidad, 67
de inc:rrmento._
posterior, 66
previo,66
de móclulo, 64
de multiplic:ac:ión, 63
de referencia, 66
de subc:onsulta especial, 302
desuma,64
de supresión de error, 70
detipo,71
de unión, 119
escalar homólogo, 119
EXISTS,302
iguales, 77
instanc:eof, 71
like,317
NOT,597
OR.:Rl
temario,69
unitario, 63
Operadores
aritm6ticos, 63
bita bit, 69
de asignac:ión, 64
de cadena, 64
de comparac:ión, 67
de matriz, 70
lógicos, 68
y su prioridad, 44
Optimizar
c:ódigo, 568
disefto, 346
tablu,347
una bue de datos, 329
Orade, 35, 37, 286
Ordenaciones
de usuario inversu, 127
definidas por el usuario, 125
Ordenar
cadenu, 153
matrices, 114
multidimensionales, 125
Ordenar matrices, 124
Otras características útiles, SS4
Otru sugerencias, 347
Otros operadores, 67
de comparación, 67

p

Piginu y sec:uenc:ias almacenadas, 434
Palabra c:lave global; 62
Pantalla del editor, 714
Papel,840
Paquetes
de datos, 440
enc:ritados, 4-40
TCP,440
Parimetros
de c:onfigurac:ión de Apache, 422
de fwrite(), 98
PDFlib,839
Pérdida o destruc:ción de datos, 389
Peri, 34, 36, 57
Pennisos,347
Personal Home Page, 33
PGP,444
Philip Greenspun, 479-
Philip R. Zimmenmnn, 444
PHP, 31, 32, 33, 34, 35, 36, 37, 38, 39, 43
ejemplo

EL_HOLISTICO@HOTMAIL.COM - 2006

add_bms.php,632
admin.php,676
AmazonResultSet.php,897
authmain.php, 548
basic_exception.php, 232
basic_stored_procedure.sql,361
bobs_front_page.php, 128
book_sc.sql, 650
bookmark_fns.php, 612
bookmarks.sql, 611
bookorama.sql,271
browserdir.php,463
calc_age.php,502
cartfunctions.php,911
change_passwd.php,624
checkout.php, 669
constants.php,891
content.html, 421
create_database.sql, 762
createauthdb.sql,415
delete_bms.php,634
delete_story.php,710
directory_submit.html, 480
directory_submit.php, 481
file_exception.php,237
filedetails.php, 466
footer.inc, 177
forgot_passwd.php,627
freight.html,82
freight.php,84
ftpmirror.php,484
header.inc, 176
home.html, 174
home.php, 175
htpass, 423
index.html, 848
index.php, 653
insert_book.php,679
inter.php, 561
iterator.php, 226
login.php, 611
logout.php, 623
lookup.php,477
make_button.php,520
member.php,620
mysql_calc_age.php,504
new_post.php,831
newbook.html, 319
page.php,697
pdf.php,857
pollsetup.sql, 527
process.php,674
processfeedback.php,142
processorder.php,100
publish.php,714
purchase.php, 669
recommend.php, 638
reflection.php,228
register_form.php,614
register_new.php,614
rejedion.html,421
resize_image.php, 700
results.php,311
results_generic.php,325
score.php,849
search.php,711
send_private_mail.php,449
services.php,219
show_book.php,658
show_cart.php,661
show_cat.php, 656
show_poll.php, 529
showpoll.php, 533
showpoll.phs,530
simplegraph.php,513
story.php, 705
upload.php, 795
user_defined_exception.php,236
vie~post.php, 829
vieworders.php,101
vieworders2.php, 131
vote.html, 528
writer.php, 703

función

addslashes(), 148
arrayO, 115
array_count_values(), 135
array_pop(), 130
array_pushO, 130
arrayJandO, 129
arrayJeVerse(),128
array...sumO,666
array_walkO, 134
arsortO, 125

asort(), 125
checkdate(), 500
chetkdnsrr(),483
chown0,468
cl~tatcache(), 468
-el~).~
copy(),469
cos(), 849
count(), 131¡
crypt(),416
current(), 134
date(),52
debug_backtrace(),235
decoct(), 467
deleteO, 749
die0,557
di(), 561
dns_get_mx(),480
doubleval0,322
each(), 118
empty(), 76
end(), 133
ereg(), 164
ereg_replace(), 165
eregiO, 164
eregi..replace(), 165
error_reporting(),598
escapeshellcmd(), 441¡
eval(), 555
exec(), 441
exit(), 548
explodeO, 132
extract(), 136
fclose(), 98
fdf_create(),857
fdf_set_file(), 857
fdf_set_value(), 857
feofO, 103
fgetcO, 105
fgetcsv(), 103
fgets(), 103
fgetss(), 103
file(), 104
file_exists(), 488
file_get_contents(), 98
file_put_contents(), 98
fileatime(),466
filegroupO,465
filemtimeO,466
fileowner(),465
filepenns(), 467
filesize(), 106
filetype(),467
flock(), 108
floor(), 503
fopen(),92
fpusthru(), 104
fputsO, 97
fread(), 106
fseekO, 107
ftellO, 107
ftp_connect(), 487
ftp_fget(),489
ftp_get(),490
ftpJogin(),487
ftp_mdtmO, 488
ftp_nlist(), 491
ftp_put(),490
ftp_quit(),487
ftp_size(), 491
func_get_arg(), 186
func_get_args(), 186
func_num_args(), 116
fwrite(),97
get_current_user(), 560
get_extension_funcs(), 559
get_loaded_extensions(), 559
get_magic_quotes_gpc(), 312
getdateO, 499
getenv(),472
gethostbyaddr(), 483
gethostbyname(), 482
getimagesize(), 797
getlastmod(), 560
gettype(),74
header(), 539
highlightJileO,562
htmlspecialchan(), 314
imap_body(), 721
imap_close(),721
imap_delete(), 721
imap_expunge(), 721
imap_fetchheader(), 721
imap_header0,721

EL_HOLISTICO@HOTMAIL.COM - 2006

imap_headers(),721
imap_open(),721
implode(), 151
ini_get(),561
ini_set(),561
intval(), 133
is_array(),75
is_callable(),75
is_dir(),467
is_double(),75
is_executable(), 467
is_file(), 4Gl
is_ftoat(), 75
is_int(),75
is_integerO,75
is_link(), 467
is_long(),75
is_null(),75
is_numeric(), 75
is_object(), 75
is_readable(),467
.is_real(), 75
is_resource(),75
is-..sadarO, -15- '
is_stringO, 75
is_uploaded.:Jile(),461
is_writable(),467
isset(), 76
join(), 151
key(),226
krsort(), 125
ksort(), 125
list(), 118
lstat(), 467
ltrim(), 144
mail(), 143
main(),588
max(), 191
md5(),416
miaotime(), SOS
mkdir(),464
mktime(),498
move_u_loaded_file(),461
mysql_error(),558
mysqli_affected_rows(),322
mysqli_connect(),315
mysqli_connect_ermo(), 312
mysqlLconnect_error(),548
mysqli_ermoO,590
mysqli_error(),590
mysqli_fetch..array(), 318
mysqli_fetchassoc()~317
mysqli_fetch_row(), 365
mysqli_num..rows(),322
mysqli_query(), 316
mysqli_select_db(),316
mysqli_stmt_bind_param(), 323
mysqli_stmt_bind_result(), 324
mysqli_stmt_execute(),323
mysqli_stmt_fetch(), 324
mysqli_stmt_prepare(),323
next(), 133
nl2br(), 144
number_format(),72
opendir(),463
passthru(),470
pdf_add_outline(), 861
pdf_begin_page(), 861
pdf_close(), 863
pdf..fillO, 870
pdf..Dew.(), 859
pdf_open(), 861
pdf_open_file(),863
pdf.:.plié:e:..iiiiige(),87o-
pdf_rect(),869
pdf_setJnfo(), 861
pdf_setlinewidth(),869
pdf_show(),863
pdf_show_xy(), 869
pdf..strolce(), 869
phpinfo(),61
pi(), 867
popen0,470
posO, 133
posix....getgrgid(),467
posix_dO,465
prev(), 133
print(), 134
print_JO,595
printfO, 145
proc_dose(),470
proc_open(), 470
putenv(),472
rangeO, 115
readdirQ,463
readfile(), 104
rename(),469
reset(), 119
rewind(),107
rmdir(), 464
rsort(), 125
rtrimO, 144
serialize(), 558
session..destroy(), 542
session..get_cookie_params(), 539
session..start(), 540
sessian_unregister(), 542
session..unset(), 542
set_error_handler(),599
set_tilne_limitO,491
setcookie(), 539
settype(), 74
shaI(),416
show_source(), 562
shuffte(), 128
sin0,871
sizeof(), 135
sort(), 114
split6;-165-
sprintfO,145
srandO, 129
stat(), 467
str_replace(),157
strcasecmp(), 153
strchr(),155
strcmp(), 153
strip_tags(), 181
stripslashes(),148
stristr(),155
strlenO,98
stmatcasecmp(), 153
stmatanp(), 153
strpol(), 156
stm:hr(),155
strrpol(), 156
strstr(), 155
strtok(), 151
strtolowerO,148
strtoupper(), 148,
substr(), 152
substr_replace(), 156
system(), 441
time(), 498
trigger_error(),599
trim(),144
uuort(), 127
ucfirst(),148
ucwordsO, 148
ubort(), 127
wnasJt0,465
unlink(),106
unserialize(), 558
unset(), 76
urlencodeO,417
usort(), 126
var_export(),228
vprintf(),147
vsprintf(),147
Hipertext Preproc:euor, 33
orientado a objetas,39, 196
versión 4 frente a versión 5,221
versión 5, 35,98
Piratas infonn6ticos, 336
Plaintext, 429
Planificar y ejecutar un proyecto de aplicad6n Wom 5lil
PNG,510
Polimorfismo,199
Politié:a de privaddad~-378
POP,438
Portabilidad,35,36,38
PostgreSQL,35,37,286
PostScript, 510
Precedenda y asodatividad,73
Presentar la solución, 608
Pretty Good Privacy,444
Principio
de asignación del privilegioaús bajo,264
de autenticad6n, 387
Privilegios, 329
actualizar, 335
Privilegios de usuario, 337
Probar
el estado de las variable, 76
y establecer tipos de variables, 74
Problemas
al abrir el archivo,96

EL_HOLISTICO@HOTMAIL.COM - 2006

con el uso de archivos planos, 109
habituales, ~1
relacionados con la Web, 338
Procedimientos almacenados, 355
Procesar
archivos,90
datos XML, 'SI
el formulario, 46
la carga de varios archivos, 795
Programación avanzada con MySQL, 354
Propiedades
de archivo, 468
handle,464
path,464
Proteger
la base de datos MySQL, 336
p6ginas móltiples, 417
Protocolo
decapa
de aplicación, 438
dered,438
de control de transmisión, 438
de Internet, '38
dered,405
de transferencia
de hipertexto seguro, 436
HTI'P /1.1, 476
IMAP,476
SMTP,476
SSL,438
Prototipos, 568
Pruebas,394
incompletas, 394

Q

Query, 33, 263

R

RAID,391
Rasmus Lerdorf, 33
Razones para utilizar PHP y MySQL, 34
RC2,400
RC4,400
RC5,400
Rt>BMS,33,37,110
Realizar
copia de seguridad de la base de datos MySQL, 348
seguimiento de las compras de un usuario mientras compra, 644
transferencia de datos inicial, 350
y anular suscripciones, 786
Recontar subexpresiones, 161
Recortar costes, 380
Recuperar
datos
con criterios espedficos, 290
con un orden dado, 297
de la base de datos, 285
desde varias tablas, 292
resultados de consulta, 317
Recursos
específicos de MySQL y SQL, 946
para Apache, 946
PHP,943
Web,942
Redes
TCP/IP,390
Reemplazos, 209
Referencias, 66
Registrar variables de sesión, SCO
Registro, 608
de transacciones, 279
Reinterpretar variables, 76
Relaciones, 249
Rendimientp, 35, 37
Reordenar matrices, 128
Repetición, 160
Representación gr6fica de datos, 509
Repudio,389
Requisitos de la solución,. 688
Resaltar c6digo fuente, SSS
Responder o reenviar correo, 751
Restablecer
contraseñas olvidadas, 626
la base de datos MySQL, 348
Reutilizar código, 168
y crear funciones, 168
Riesgos y amenazas, 371
RTF es un formato de texto, 842

S

Saber cúndo parar
feof(), 103
SAMBA,406
Sangrado, 573
Secuencia de comandos
add_bnúorm.php, 631
add_bms.php, 632
book_sc.sql, 651
CGI,930
c:hange_passwd.php, 624
change_passwd_form.php,624
como argumento, 563
con el nombre
resiz.e_ image.php, 700
de carga, 759
de configuración, 720
de exploración de directorios, 463
dePHP,46
de servidor, 33, 44
del nócleo, 51
delete_bms.php, 634
edit_booltJorm.php, 680
filedetails.php, 465
forgot_form.php, 627
forgot_passwd.php, 6'Zl
go-pear, 939
index.php, 652
insert_book.php,679
keyword_add.php, 711
keywords.php, 711
logout.php, 551
make_button.php, 520
member.php, 620
populate.sql, 725
process.php, 673
publish.php, 714
publish_ story.php, 715
purchase.php, 669
register_new.php, 614
results.php, 313
score.php, 849
show_book.php, 658
show_cart.php, 65
show _cat.php, 652
view _post.php, 822
vieworders.php, 102
Seguridad
ffsica, 388
Seleccionar
claves lógicas, 253
modos de archivo, 91
tipos de dato de columna, '171
un entorno de desarrolló; 568
una base de datos, 309
una cuenta, 738
una estrategia, :171
Separar lógica y contenido, 578
Serialización, 555
Servicios Web, 33, 876
con XML y SOAP, 33
Servidor
multiusuario, 33
secundario, 349
Web, 33, 34, 44
Web seguro, 387
Sesión
iniciar, 261
Iniciar en el servidoi-FrP, W1
iniciar en MySQL, 261
iniciar y cerrar, 730
SHA·l,416
S-HTI'P, 436
Shutdown, 331
SimpleXML, 37, 875
Sistema de privilegios de MySQL, 263
Sistemas, 407
BSD,446
de administración de contenido, 32
de compresión, 440
de distribución inmediata, 380
de encriptación, 396
de permisos, 347
de preguntas y respuestas, 844
existentes, 688
$L,378
Sistemas de administración de base de datos, 109
Slashdot, 410
SMTP,438
SOAP,872

EL_HOLISTICO@HOTMAIL.COM - 2006

Sobrecargar IMtodos, 224
Software
de generación de documentos, 844
para crear
PDF mediante
programación, 8,16
Solaris, 36, 950
Solucionar
errores con elegancia, 585
problemas con los encabezados, 871
SORT_NUMERIC, 124
SORT...;REGULAR, 124
SORT_STRING, 124
S(IL,33,35,37,39,111
add,306
ali, 220
alter, 331
ALTERTABLE,304
and,68
any,635
are, 106
as,37, 55
ase, 298
at.51
avg, 299
egin,363
between, 291
bit, 69
boolean, 76
by,80
call, 224
case, 80
char, 105
check. 223
clase, 312
column, 121
CIDIIIIInit, 358
CIIIIIMct. 312
CIIIAl.... , 262
CIDIIIIII'aild. 360
mnn 16
UD
crea•11t
CREAD D"'TAMSE. 341
CREATE INDEX. 276
CREATETAIILE.267
cunent. 11_7
cunor,363
DATABASE,307
decimal, 146
declare,87
DECLARE, 363
delete, 106
DELETE;266
desc,298
describe, 92
DESCRIBE, 275
disconnec:t, 326
distinct, 637
double,60_
drop,269
DROP DATABASE, 307
DROP INDEX, 305-
DROP TABLE, 307
else, 44
end, 133
escape, 93
exception, 232
exec,441
exec:ute, 323
exists.106
explain, 340
externa), 889
extract, 136
false, 67
fetch, 312
first, 927
Ooat, 76
for,44
foreign, 360
found,81
from, 143
full, 694
function, 126
get, 46
global,62
go,312
grant, 261
GRANT,264
group,299
having,300
in,67
initially, 927
inner, 865

input,45
insert, 269
INSERT,266
int, 76
integer, 901
into, 286
is,55
join, 151
key, 118
last, 155
left, 174
like, so
LOAD,350
local, 72
LOCK TABLE, 267
lower, 160
ltrim, 144
match,615
max, 191
min, 163
natural,153
next, 133
no, 31, 32, 33, 34, 36, 37, 38, 39, 43
not, 77
null, 271
of,55
on, 331:34, 35, 37, 39, 44
only,459
open,49
OPEN,341
option,80
or,43
order,50
out, 70
output,416
pad, 128
prepare, 323
primary, 271
privileges, 269
procedure,361
public,96
Ptrr,457
read,350
real, 32, 39, 40, 43
references, 360
revolee, 269
REVOKE,264
right,83
rollback. 358
rows,275
rtrim.144
select, 80
SELECT,266
set, 203
SET, 108
some, 175
space, 160
sql,271
sqlstate~ 364
substr, 152
sum,361
table,45
temporary, 345
tinyint, 272
to,50
transaction, 358
trim, 144
true,67
unique,344
unknown,467
update, 269
UPI>ATE;266
upper, 160
user,265
using, 295'
values, 135
vuchar, 273
view,413
where,290
with, 146
write,693
year,499
S(lUte, 35, 37; 111
SSL96
Steve Muanda, 535
Subcansultas
búcas.300
de&las.302
relacionadas, 302
y operadores, 301
Subexpresiones, 160
Suministrar
servicios y artículos digitales. m
transacciones 433

EL_HOLISTICO@HOTMAIL.COM - 2006

ciones enóneas hechas por los desarrolladores, 394
iir,subcadenas, 156
expresiones regulares, 165
:, 35, 28\ -

umns_priv, 334
330
t, 330
es_priv, 330
r, 330
, 246
o de un archivo, 106
390
DP, 438
as avanzadas de PHP, 39, 453
e 395
38, 58
cadena, 280
columna, 273
combinación, 296
datos de PHP, 59
fecha y hora, 279
sitios Web comerciales, 371
tablas, 254
variables, 58
ME,458
méricos, 278
iveles de privilegio, 266
de decisiones, 44
estructuras é9ndicionales 77
ir con la base de datos de' MySQL, 284
ıcciones 355
'erencia '.
hiv0& de texto, 489.
erencia FTP, 489
DES, 400

rmidad: 17-0
rsity of Maryland, -W6
33, 34, 36, 38, 96

comando
mget,490
rcp,389
touch, 468
traceroute, 390
URL,46
Usuario nobody, 446
Utilizar
__autoload(), 225
alias, 296
array_reverse(), 129
atributos de clase, 203
autenticación, 387
autenticación básica enPHP,419¡
auto_prepend_file y auto_append_file, 178
bucles, 116
para acceder a la matriz, 116
clases abstractas, 224
comillas mágicas, 555
constantes de clase, 222
cookies con sesiones, 539
declare, 87
el API de reflexión, 228
el control de sesiones en PHP, 536
el monitor de MySQL, 261
encriptadón en PHP, 433
estos elementos en Smart Form 163
etiquetas PHP, 48
explode(), implode() y join(), 150
fopen() para abrir un archivo, 92
formatQ HTML, 144
FTP,475
para realizar una copia, 484
funciones
de búsqueda
dered,475
de calendario, 495
de ejecución de programas, 469
de red y de protocolo, 474
de variables, 44
imágenes generadas automáticamente en otras páginas,509
include(), 177

case 'reply'

indices, 347
instrucciones predefinidas, 3~0
la base de datos correcta, 262
la clase treenode, 822
la función date(), 52
la función getdateO, 499
la secuencia de comandos show.cart.php, 659
las funciones de directorio, 462
las funciones de PHP, 179
las interfaces de servicios Web de mazon, 882
matrices, 112
metadatos, 690
microsegundos, 505
mod_auth_mysql, 428
MySQL. 39, 243
operadores, 63
calcular los totales de los formularios, 71
otras funciones de Fl'P, 491
otras interfá~ de base de datos y PHP,324
otros sitios Web, 475
parámetros, ~69
PEARDB,324
PHP, 39, 41, 44
en la linea de comandos, 555
y MySQL en grandes proyectos, 566
productos Zend, 580
require(), 169
e include(), 169
para plantillas de sitios Web, 173
equire_once() e include_once(), 178
REST/XML sobre HTTP,902
shuffle(), 128
SOAP,883
SOAP con PHP, 883
sort(), 124
SSL,433
strtok(), 151"
subconsultas, 285
substr(), 152
texto y fuentes para crear imágenes, 509
transacciones con InnoDB, 358
un sistema existente; 684
una estructura de directorios, estándar, 574
una optimización sencilla, 579
una subconsulta como tabla temporal,303
valores
de columna atómicos, 252
predeterminados, 347
variables de sesión, 540

V

Validar fechas,500
Variable
$_SESSION, 542
$_SESS10N['sess_var'), 5424
$_SESSION['valid_user'), ióO
Saction, 764
Sbutton_text, 522
Schar, 105
$color, 522
Scurrent, 117
Selement, 118
Sfp,97
Shost,486
Slocalfile, 486
Slocaltime, 48J
$m_childlist, 1125
Sm_depth,,825
$name,85é
Snothere, 76
$output, 855
Sremotefile, 486
Sremotetime, 488
Sresult,64
Ssearchtype,314
$tireqty, 54
$total, 59
Stotal_shipping, 146
$totalamount, 59
$value, 189
$var, 187
Swidth,531
account, 731
action,730

EL_HOLISTICO@HOTMAIL.COM - 2006

browseNode, 892
cart, 893
corto
$DOCUMENT_ROOT,101
$email, 615
$isbn,321
$new_url,632
$old..passwd, 625
$tireqty, 100
$wiername, 6'D
datadir, 'J:/6
ele entorno GDFONTPAnl,523
de PHP, 53
de sesi6ñ,541
$selected_account,741
admin_user,678
auth_aser, 704
cart, 659
expanded, 819
valid_user, 549
de !ipo, 59
entero, 59
flotante, 322
espedai llamada $this, 203
$_5ESSION, 538
$php_errormsg, 70
incorporada
SHI'IPJBM!ll.V~
..ROOT'J,92
PHP, 558
POST; **667**
superglobal
$..REQ1$1', 522
$_5ER\.'BR¡ 21JIL
Variables
cortos
$de1Jne, 634
$name,669
Sql,849
$seuditype,312
$tireqty, 55
$usemame,620
ele cooldes,62
de entorno, 62
de formulario,53
dePHP,74
de servidorpredefinidas,9;3
de tipo, 60
de tipo variable,60
del archivo php.ini, 561
del sistema MySQL,341
escalares, 71
globales, 54
locales,363
miembro de- la clue, 825
POST,595
superglobales, 62
y literales, 57
Ver
el carro, 663
encabezados ele mensaje,747
información ele listas,782
los contenidos del buzón
VeriSign, 395

W

WBMP,511
Windows,44
Windowa NT, 44
WSDL,880

X

XML,876

Z

Zend Technologies,33, 35, ~. 58Ó

EL_HOLISTICO@HOTMAIL.COM - 2006