

L A B I B L I A



Mohammed J. Kabir

# Servidor Apache 2

INCLUYE  
CD-ROM



WILEY

**ANAYA**  
MULTIMEDIA

**La biblia de**

# **Servidor Apache 2**

# LA BIBLIA DE

TÍTULO DE LA OBRA ORIGINAL:  
Apache Server 2 Bible

RESPONSABLE EDITORIAL:  
Víctor Manuel Ruiz Calderón  
Susana Krahe Pérez-Rubín

TRADUCTOR:  
Seven Servicios Integrales

AUTOEDICIÓN:  
Seven Servicios Integrales

**La biblia de**

# **Servidor Apache 2**

**Mohammed J. Kabir**



Todos los nombres propios de programas, sistemas operativos, equipos hardware, etc. que aparecen en este libro son marcas registradas de sus respectivas compañías u organizaciones.

Reservados todos los derechos. El contenido de esta obra está protegido por la ley, que establece penas de prisión y/o multas, además de las correspondientes indemnizaciones por daños y perjuicios, para quienes reprodujeren, plagiaren, distribuyeren o comunicasen públicamente, en todo o en parte, una obra literaria, artística o científica, o su transformación, interpretación o ejecución artística fijada en cualquier tipo de soporte o comunicada a través de cualquier medio, sin la preceptiva autorización.

Copyright © 2002 by Anaya Multimedia.

Original English language edition copyright © 2002 by Hungry Minds, Inc.

All rights reserved including the right of reproduction in whole or in part in any form. This edition published by arrangement with the original publisher, Hungry Minds, Inc.

Edición española:

© EDICIONES ANAYA MULTIMEDIA (GRUPO ANAYA, S.A.), 2003

Juan Ignacio Luca de Tena, 15. 28027 Madrid

Depósito legal: M. 51.355-2002

ISBN: 84-415-1468-2

Printed in Spain

Imprime: Lavel, S. A.

*A la memoria de mi madre,  
Nazma Bathen.*

# Agradecimientos

Me gustaría dar las gracias al grupo Apache por haber creado el servidor Web más poderoso, ampliable y modular del mundo. Quiero darle especialmente las gracias a Ralf S. Engelschall. Ralf, es el autor del módulo `mod_rewrite`, y proporcionó un gran soporte al desarrollo del capítulo 9 sobre las reglas para reescribir las URL. Los ejemplos prácticos de ese capítulo pertenecen a su colección personal, que mantiene en aumento su sitio Web [www.engelschall.com/pw/apache/rewriteguide](http://www.engelschall.com/pw/apache/rewriteguide).

Quiero dar las gracias también al equipo de Hungry Minds, el cual hizo de este libro una realidad. Es imposible realizar una lista con todos los implicados pero debo mencionar a las siguientes personas:

James Russell, el Director de desarrollo del proyecto, el cual empujó este proyecto. No sé cómo hubiese podido hacer este libro sin su generosa ayuda y sus sugerencias a cada paso del camino. Gracias James.

Terri Varveris, el Director de adquisiciones, que me ofreció esta oportunidad y que aseguró su final. Gracias Terri.

Sheila Kabir, mi esposa, que tuvo que cargar con muchas horas de trabajo duro durante los meses que tardé en escribir este libro. Gracias cielo.

## Acerca del autor

Mohammed Kabir es el fundador y el director general de Evoknow, Inc. Esta compañía está especializada en el desarrollo de software CRM. Kabir disfruta viajando, siempre y cuando no esté ocupado gestionando proyectos o escribiendo libros. Estudia ingeniería de sistemas computacionales en la Universidad estatal de California, en Sacramento. Se puede contactar con él en [kabir@evoknow.com](mailto:kabir@evoknow.com).

# Índice

Agradecimientos .....	6
Acerca del autor .....	6
<b>Introducción .....</b>	<b>33</b>
Cómo está organizado este libro .....	34
Parte I. Comenzar .....	34
Parte II. Administrar sitios Web .....	34
Parte III. Ejecutar aplicaciones Web .....	34
Parte IV. Asegurar su sitio Web .....	34
Parte V. Ejecutar Apache en Win32 .....	34
Parte VI. Mejorar la escalabilidad.....	35
Parte VII. Apéndices .....	35
Convenios utilizados en este libro .....	35
<b>Parte I. Comenzar .....</b>	<b>37</b>
<b>1. Apache: el servidor número uno .....</b>	<b>39</b>
En este capítulo .....	39
Popularidad de Apache .....	40
Apache: el comienzo .....	41
La lista de características de Apache .....	41
Entender la arquitectura de Apache 2.0.....	44
Módulos multiproceso .....	44
El MPM prefork .....	44

El MPM threaded .....	44
El MPM perchild .....	45
El MPM winnt .....	45
Filtrado I/O .....	45
El nuevo demonio CGI .....	46
Apache es portable en tiempo de ejecución .....	46
Entender la licencia de Apache .....	47
<b>2. Obtener e instalar Apache .....</b>	<b>51</b>
En este capítulo .....	51
La fuente oficial de Apache .....	52
Requisitos del sistema .....	52
Requisitos para construir Apache desde la distribución de la fuente .....	53
Requisitos para ejecutar un servidor Web Apache .....	54
Bajar el software .....	56
Instalar Apache desde el código fuente .....	57
Configurar la fuente de Apache .....	58
Opciones avanzadas de configuración para sitios con mucho tráfico .....	63
Compilar e instalar Apache .....	65
Instalar Apache desde los paquetes binarios RPM .....	69
Mantenerse al día en el desarrollo de Apache .....	69
<b>3. Preparar y ejecutar Apache .....</b>	<b>73</b>
En este capítulo .....	73
Configurar Apache .....	73
Configurar el entorno global para Apache .....	78
PidFile .....	79
ScoreBoardFile .....	80
Timeout, KeepAlive, MaxKeepAliveRequests y KeepAliveTimeout ....	80
Contenedores IfModule .....	80
Directivas para el comportamiento MPM threaded (comportamiento MPM por defecto) .....	81
StartServers .....	81
MaxClients .....	81
MinSpareThreads .....	82
MaxSpareThreads .....	82
ThreadsPerChild .....	82
MaxRequestPerChild .....	82
Configurar el servidor principal .....	83
Puerto .....	83
Directivas de usuarios y grupos .....	83
ServerAdmin .....	84
DocumentRoot .....	84

Directivas en contenedores de directorios .....	86
UserDir .....	88
DirectoryIndex .....	89
AccessFileName .....	91
Contenedor de archivos .....	91
UseCanonicalName .....	91
TypesConfig .....	91
DefaultType .....	91
Contenedor IfModule .....	91
HostnameLookups .....	92
ErrorLog .....	92
LogLevel .....	92
CustomLog .....	92
ServerSignature .....	93
Alias .....	93
ScriptAlias .....	93
El resto de directivas .....	94
LanguagePriority .....	94
AddDefaultCharset .....	94
Iniciar y parar Apache .....	94
Iniciar Apache .....	95
Reiniciar Apache .....	97
Parar Apache .....	97
Parar Apache automáticamente .....	97
Parar el servidor Apache manualmente .....	97
Comprobar Apache .....	98
<b>4. Configurar Apache con directivas MPM Winnt.....</b>	<b>101</b>
En este capítulo .....	101
Contextos de las directivas Apache .....	102
Contexto de configuración del servidor .....	103
Contexto de contenedor .....	103
Contexto en el ámbito de directorio .....	105
Directivas de configuración general .....	106
AccessFileName .....	106
AddDefaultCharset .....	107
ContentDigest .....	107
DefaultType .....	108
DocumentRoot .....	108
ErrorDocument .....	109
<IfDefine> .....	111
<IfModule> .....	111
Include .....	112

Options .....	113
Port .....	115
ServerAdmin .....	115
ServerName .....	116
ServerRoot .....	117
ServerSignature .....	117
ServerTokens .....	117
SetInputFilter .....	118
SetOutputFilter .....	118
<b>Directivas de rendimiento y de configuración de recursos .....</b>	<b>118</b>
Controlar los procesos de Apache .....	119
ListenBacklog .....	119
MaxClients .....	119
MaxRequestsPerChild .....	119
MaxSpareServers .....	119
MinSpareServers .....	120
SendBufferSize .....	120
StartServers .....	120
TimeOut .....	120
Realizar conexiones persistentes .....	120
KeepAlive .....	121
KeepAliveTimeout .....	121
MaxKeepAliveRequests .....	122
Controlar los recursos del sistema .....	122
RLimitCPU .....	122
RLimitMEM .....	123
RLimitNPROC .....	123
UseCanonicalName .....	124
Utilizar módulos dinámicos .....	124
AddModule .....	124
ClearModuleList .....	125
<b>Directivas de contenedores estándar .....</b>	<b>125</b>
<Directory> .....	125
<DirectoryMatch> .....	127
<Files> .....	127
<FilesMatch> .....	128
<Location> .....	128
<LocationMatch> .....	129
<b>Directivas específicas de host virtuales .....</b>	<b>129</b>
NameVirtualHost .....	129
ServerAlias .....	130
ServerPath .....	131
<VirtualHost> .....	131

Directivas de registro .....	132
LogLevel .....	133
PidFile .....	134
ScoreBoardFile .....	134
Directivas de autentificación y de seguridad .....	135
AllowOverride .....	135
AuthName .....	136
AuthType .....	137
HostNameLookups .....	137
IdentityCheck .....	138
<Limit> .....	138
<LimitExcept> .....	139
LimitRequestBody .....	139
LimitRequestFields .....	140
LimitRequestFieldsize .....	140
LimitRequestLine .....	140
Require .....	140
Satisfy .....	141
ScriptInterpreterSource .....	142
Directivas específicas de MPM threaded .....	142
CoreDumpDirectory .....	143
Group .....	143
Listen .....	143
ListenBacklog .....	144
LockFile .....	144
MaxClients .....	145
MaxRequestsPerChild .....	145
MaxSpareThreads .....	146
MinSpareThreads .....	146
SendBufferSize .....	147
StartServers .....	147
ThreadsPerChild .....	148
User .....	148
Directiva específicas de MPM perchild .....	149
AssignUserID .....	149
ChildPerUserID .....	150
ConnectionStatus .....	150
CoreDumpDirectory .....	151
Group .....	151
Listen .....	151
ListenBacklog .....	151
LockFile .....	151
MaxRequestsPerChild .....	151

MaxSpareThreads .....	151
MaxThreadsPerChild .....	151
MinSpareThreads .....	152
NumServers .....	152
PidFile .....	152
ScoreBoardFile .....	152
SendBufferSize .....	152
StartThreads .....	153
User .....	153
<b>Directivas específicas de MPM .....</b>	<b>153</b>
CoreDumpDirectory .....	153
Listen .....	153
ListenBacklog .....	153
MaxRequestsPerChild .....	153
PidFile .....	154
SendBufferSize .....	154
ThreadsPerChild .....	154
<b>Directivas específicas de MPM prefork .....</b>	<b>154</b>
CoreDumpDirectory .....	154
Group .....	154
Listen .....	154
ListenBacklog .....	154
LockFile .....	155
MaxClients .....	155
MaxRequestsPerChild .....	155
MaxSpareServers .....	155
MinSpareServers .....	155
PidFile .....	156
ScoreBoardFile .....	156
SendBufferSize .....	156
StartServers .....	156
User .....	156
<b>5. Módulos Apache .....</b>	<b>159</b>
En este capítulo .....	159
Un vistazo a los módulos .....	160
Módulos relacionados con el entorno .....	160
mod_env .....	161
PassEnv .....	161
SetEnv .....	161
UnsetEnv .....	162
mod_setenvif .....	162
BrowserMatch .....	162

BrowserMatchNoCase .....	163
SetEnvIf .....	163
SetEnvIfNoCase .....	163
mod_unique_id .....	164
Módulos de control de acceso y autentificación .....	164
mod_auth_anon .....	165
Anonymous .....	165
Anonymous_Authoritative .....	165
Anonymous_LogEmail .....	166
Anonymous_MustGiveEmail .....	166
Anonymous_NoUserID .....	166
Anonymous_VerifyEmail .....	167
mod_auth_dbm .....	167
AuthDBMUserFile .....	170
AuthDbmGroupFile .....	170
AuthDBMAuthoritative .....	171
mod_auth_db .....	172
AuthDBUserFile .....	173
AuthDBGGroupFile .....	173
AuthDBAAuthoritative .....	174
Módulos de generación de contenido dinámico .....	174
mod_actions .....	175
Action .....	175
Script .....	178
mod_ext_filter .....	179
ExtFilterDefine .....	179
ExtFilterOptions .....	180
Módulos de configuración de tipo de contenido .....	181
mod_mime .....	181
AddCharset .....	181
AddEncoding .....	182
AddHandler .....	182
AddLanguage .....	182
AddType .....	183
DefaultLanguage .....	183
ForceType .....	184
SetHandler .....	184
RemoveHandler .....	184
TypesConfig .....	185
mod_mime_magic .....	185
mod_negotiation .....	186
CacheNegotiatedDocs .....	186
LanguagePriority .....	187

Módulos de listas de directorios .....	187
mod_dir .....	188
mod_autoindex .....	188
AddAlt.....	189
AddAltByEncoding .....	189
AddAltByType .....	190
AddDescription .....	190
AddIcon .....	190
AddIconByEncoding .....	191
AddIconByType .....	191
DefaultIcon .....	192
FancyIndexing .....	192
HeaderName .....	192
IndexIgnore .....	193
IndexOptions .....	193
IndexOrderDefault .....	195
ReadmeName .....	195
Response Header Modules .....	195
mod_asis .....	196
mod_headers .....	197
mod_expires .....	197
ExpiresActive .....	198
ExpiresByType .....	198
ExpiresDefault .....	199
mod_cern_meta .....	200
MetaFiles .....	200
MetaDir .....	200
MetaSuffix .....	201
Módulos de información de servidores y de registro .....	201
mod_log_config .....	202
mod_status .....	202
mod_info .....	202
mod_usertrack .....	202
Módulos de integración URL .....	202
mod_userdir .....	203
mod_alias .....	204
Alias .....	204
AliasMatch .....	205
Redirect .....	205
RedirectMatch .....	206
RedirectTemp .....	207
RedirectPermanent .....	207
ScriptAlias .....	207

ScriptAliasMatch .....	207
mod_speling .....	208
mod_vhost_alias .....	208
VirtualDocumentRoot .....	208
VirtualDocumentRootIP .....	209
VirtualScriptAlias .....	210
VirtualScriptAliasIP .....	210
Otros módulos .....	210
mod_so .....	211
LoadFile .....	211
LoadModule .....	211
mod_imap .....	212
ImapMenu .....	213
ImapDefault .....	214
ImapBase .....	215
mod_file_cache .....	215
MMapFile .....	215
CacheFile .....	216
mod_dav .....	216
Dav .....	216
DavLockDB .....	216
DavMinTimeout .....	216
DavDepthInfinity .....	217
<b>Parte II. Administrar sitios Web .....</b>	<b>219</b>
<b>6. Alojar sitios Web virtuales .....</b>	<b>221</b>
En este capítulo .....	221
Entender las capacidades del hospedaje virtual en Apache .....	222
Establecer un host virtual .....	223
Host virtuales basados en nombre .....	224
Host virtuales basados en IP .....	225
Varios servidores principales como host virtuales .....	226
Configurar DNS para un host virtual .....	229
Entender los archivos de zona .....	229
Establecer las DNS para host virtuales nuevos .....	231
Ofrecer servicios de correo virtual .....	231
Asignar usuario y grupo a cada host virtual .....	232
Gestionar un gran número de host virtuales .....	234
Configuración automática de host virtuales utilizando mod_perl .....	235
Generar la configuración de host virtuales utilizando el script makesite .....	238
Gestionar host virtuales utilizando MySQL con el módulo mod_v2h .....	242

<b>7. Autentificación y autorización de visitantes al sitio Web .....</b>	<b>245</b>
En este capítulo .....	245
Autentificación vs. autorización .....	246
Entender cómo funciona la autentificación .....	246
Autenticar usuarios mediante el módulo mod_auth .....	248
Entender las directivas mod_auth .....	249
Directiva AuthUserFile .....	249
Directiva AuthGroupFile .....	249
Directiva AuthAuthoritative .....	250
Crear una sección sólo de miembros en su sitio Web .....	251
Crear una sección sólo de miembros utilizando un archivo .htaccess .....	252
Agrupar usuarios para accesos restringidos a distintas secciones Web .....	254
Autorizar el acceso mediante el nombre del host o las direcciones IP .....	256
Directiva allow .....	256
Directiva deny .....	257
Directiva order .....	258
Directiva allow from env=variable .....	258
deny from env=variable .....	259
Combinar autentificación y autorización .....	260
Autentificación con bases de datos relacionales .....	261
Utilizar un servidor con una base de datos MySQL para la autentificación .....	262
Crear la base de datos de autentificación de usuarios en el servidor MySQL .....	262
Conceder acceso al servidor Apache a la base de datos de autentificación de usuarios en MySQL .....	265
Compilar e instalar el módulo mod_auth_mysql .....	266
Autenticar usuarios utilizando el módulo mod_auth_mysql .....	267
Utilizar otras bases de datos para autentificación de usuarios .....	269
Gestionar usuarios y grupos en una RDBM .....	271
Utilizar cookies para autenticar sesiones .....	276
<b>8. Monitorización del acceso a Apache.....</b>	<b>283</b>
En este capítulo .....	283
Monitorizar Apache .....	284
Acceder a la información de configuración con mod_info .....	284
Permitir páginas de estado con mod_status .....	287
Ver páginas de estado .....	288
Simplificar el despliegue de estado .....	290
Almacenar información del estado del servidor .....	291
Crear archivos de registro .....	292
Directiva TransferLog .....	293
Directiva LogFormat .....	294

Directiva CustomLog .....	295
Directiva CookieLog .....	296
Personalizar sus archivos de registro .....	296
Crear varios archivos de registro .....	299
Registrar cookies .....	300
Directiva CookieExpires .....	301
Directiva CookieTracking .....	302
Utilizar registros de error .....	302
Analizar sus archivos de registro .....	304
Mantenimiento de registros .....	306
Utilizar rotatelog .....	306
Utilizar logrotate .....	307
Utilizar logresolve .....	308
<b>9. Reescribir las URL .....</b>	<b>313</b>
En este capítulo .....	313
El motor de reescritura de URL de Apache .....	314
RewriteEngine .....	316
RewriteOptions .....	317
RewriteRule .....	318
RewriteCond .....	321
RewriteMap .....	323
RewriteBase .....	325
RewriteLog .....	325
RewriteLogLevel .....	325
RewriteLock .....	326
Distribución de las URL .....	326
Ampliar una URL a la forma canónica de las URL .....	326
Redirigir un directorio home de usuario a un nuevo servidor Web .....	328
Buscar una página en varios directorios .....	329
Asignar una variable de entorno basándose en una URL .....	332
Crear sitios www.username.domain.com .....	333
Redireccionar una URL fallida a otro servidor Web .....	335
Crear un acceso multiplexor .....	335
Crear URL dependientes del tiempo .....	337
Manejar contenido .....	338
Añadir compatibilidad retroactiva en las URL .....	338
Crear las URL con contenido específico para el navegador .....	338
Crear HTML para un puente CGI .....	339
Restricción de acceso .....	339
Robots de bloqueo .....	340
Crear deflector URL basado en una referencia HTTP .....	340

<b>10. Establecer un servidor Proxy .....</b>	<b>343</b>
En este capítulo .....	343
¿Quién debería utilizar un servidor proxy? .....	344
Análisis de los tipos de servidores proxy .....	344
Proxy forward .....	345
Proxy reverse .....	346
Directivas mod_proxy .....	347
ProxyRequests .....	347
ProxyRemote .....	347
ProxyPass .....	348
ProxyBlock .....	349
NoProxy .....	349
ProxyDomain .....	350
CacheRoot .....	350
CacheSize .....	351
CacheGcInterval .....	351
CacheMaxExpire .....	351
CacheLastModifiedFactor .....	352
CacheDirLength .....	352
CacheDirLevels .....	353
CacheDefaultExpire .....	353
NoCache .....	353
Configurar un servidor proxy Apache .....	354
Escenario 1: conectar una IP privada a Internet .....	355
Escenario 2: caching sitios web remotos .....	355
Escenario 3: crear una copia local de un sitio Web .....	357
Preparar un navegador Web para utilizar un proxy .....	357
Configuración manual del proxy .....	358
Configurar Netscape manualmente .....	358
Configurar Internet Explorer manualmente .....	359
Configuración automática del proxy .....	360
Asignar valores de retorno para FindProxyForURL .....	362
Utilizar funciones predefinidas en FindProxyForURL .....	363
Escenario 1: utilizar un proxy únicamente para solicitudes	
URL remotas .....	365
Escenario 2: utilizar varios servidores proxy .....	367
Escenario 3: generar FindProxyForURL dinámicamente utilizando	
un script CGI .....	369
<b>11. Ejecutar sitios Web perfectos .....</b>	<b>373</b>
En este capítulo .....	373
Ciclo de desarrollo Web .....	374
Poner en marcha el ciclo Web .....	376

Establecer el ciclo Web .....	377
Crear un host virtual para cada fase .....	378
Utilizar varios procesos (principales) del servidor Apache .....	379
Utilizar varios ordenadores servidores Apache para el ciclo Web .....	381
Implementar el ciclo Web .....	381
Probar el ciclo Web .....	382
Mover el sitio nuevo al servidor de producción .....	382
Construir un sitio Web utilizando plantillas y el makepage .....	384
Utilizar HTTP PUT para publicaciones Web en una Intranet .....	387
Las directivas del módulo mod_put .....	387
EnablePut .....	387
EnableDelete .....	387
umask .....	387
Compilar e instalar mod_put .....	388
Establecer un directorio Web que permita el método PUT .....	388
Establecer un host virtual para utilizar el módulo mod_put .....	390
Mantenimiento de su sitio Web .....	392
Backup online .....	392
Backup offline .....	393
Estandarizar estándar .....	393
Política de desarrollo de documentos HTML .....	394
Utilice siempre etiquetas HTML estándar .....	394
Guarde imágenes in-line junto con los documentos .....	394
Desplegar mensajes copyright en cada documento .....	396
Política de desarrollo de aplicaciones dinámicas .....	396
Utilice siempre un control de la versión .....	396
No utilice nombres de rutas absolutos en los scripts ni en las aplicaciones .....	397
Proporcionar documentación de usuario y de código .....	397
Evitar las etiquetas HTML embebidas en scripts o en aplicaciones .....	397
No confiar en los datos introducidos por el cliente .....	397
Evitar las variables globales en los scripts CGI basados en Perl .....	397
Proporcionar a su sitio Web una interfaz intuitiva .....	398
Facilite la navegación en su sitio .....	398
Crear un diseño atractivo .....	399
Colores apropiados .....	399
Tamaño apropiado de texto .....	399
Mínima utilización de imágenes y animaciones .....	399
Elimine los mensajes de error en clave .....	400
Pruebe su GUI Web .....	400
Promocionar su sitio Web .....	401

<b>Parte III. Ejecutar aplicaciones Web .....</b>	<b>403</b>
<b>12. Ejecutar scripts CGI .....</b>	<b>405</b>
En este capítulo .....	405
¿Qué es CGI? .....	406
Input y Output CGI .....	407
Solicitudes GET .....	407
Solicitudes POST .....	410
Comparar GET y POST .....	411
Decodificación de los datos introducidos .....	412
Variables CGI Apache .....	413
Variables del servidor .....	413
SERVER_SOFTWARE .....	414
SERVER_ADMIN .....	414
DOCUMENT_ROOT .....	414
Variables para las solicitudes del cliente .....	414
SERVER_NAME .....	415
HTTP_HOST .....	415
HTTP_ACCEPT .....	415
HTTP_ACCEPT_CHARSET .....	415
HTTP_ACCEPT_ENCODING .....	415
HTTP_ACCEPT_LANGUAGE .....	416
HTTP_USER_AGENT .....	416
HTTP_REFERER .....	416
HTTP_CONNECTION .....	417
SERVER_PORT .....	417
REMOTE_HOST .....	417
REMOTE_PORT .....	417
REMOTE_ADDR .....	417
REMOTE_USER .....	418
SERVER_PROTOCOL .....	418
REQUEST_METHOD .....	418
REQUEST_URI .....	418
REMOTE_IDENT .....	418
AUTH_TYPE .....	419
CONTENT_TYPE .....	419
CONTENT_LENGTH .....	419
SCRIPT_NAME .....	419
SCRIPT_FILENAME .....	419
QUERY_STRING .....	419
PATH_INFO .....	420
PATH_TRANSLATED .....	420
Configurar Apache para CGI .....	420

Análisis del directorio de programas CGI .....	421
Elegir extensiones específicas de archivos CGI .....	422
Permitir el acceso cgi-bin a sus usuarios .....	424
Contenedores Directory o DirectoryMatch .....	424
ScriptAliasMatch.....	425
Crear nuevas extensiones CGI utilizando AddType .....	427
Ejecutar programas CGI .....	427
Escribir scripts CGI en Perl .....	428
Análisis de un script CGI sencillo .....	430
Crear un procesador básico de formularios Web .....	435
Permitir soporte de depuración de errores CGI en Apache .....	457
ScriptLog.....	457
ScriptLogLength .....	458
ScriptLogBuffer .....	458
Depurar errores en sus scripts basados en Perl .....	458
Depuración de errores desde la línea de comandos .....	458
Depuración utilizando la impresión de registros y de depuración.....	460
Depurar con CGI::Debug .....	462
<b>13. Server Side Includes (SSI).....</b>	<b>467</b>
En este capítulo .....	467
Server Side Include .....	468
Configurar Apache para SSI .....	469
Activar SSI para un directorio completo .....	469
Activar SSI para un tipo específico de archivo .....	470
Utilizar XBitHack para archivos .htm o .html .....	471
Utilizar comandos SSI .....	472
config .....	473
echo .....	476
exec .....	476
fsize .....	481
flastmod .....	481
include .....	482
printenv .....	483
set .....	483
Variables SSI .....	483
Control de flujo de los comandos .....	484
<b>14. Configurar Apache para FastCGI.....</b>	<b>489</b>
En este capítulo .....	489
FastCGI .....	489
Alcanzar alto rendimiento utilizando caching .....	491
Escalabilidad a través de aplicaciones distribuidas.....	492

Entender cómo funciona FastCGI .....	494
Arquitectura básica de una aplicación FastCGI .....	497
Distintos tipos de aplicaciones FastCGI .....	498
Migración desde CGI a FastCGI .....	499
Puntos que hay que recordar sobre la migración .....	500
Un ejemplo de un script de migración .....	501
Establecer FastCGI en Apache .....	504
Directivas FastCGI para Apache .....	504
Directiva AppClass .....	505
Directiva ExternalAppClass .....	506
Directiva FastCgiIpcDir .....	507
Configurar httpd.conf para FastCGI .....	507
<b>15. PHP y Apache .....</b>	<b>513</b>
En este capítulo .....	513
Entender cómo funciona PHP .....	514
PHP en su compañía .....	515
Requisitos previos para PHP .....	517
Compilar e instalar PHP .....	517
Construir PHP como una solución CGI .....	518
Construir PHP como un módulo Apache .....	518
Construir PHP como un módulo estático de Apache .....	518
Construir PHP como un módulo Dynamic Shared Object (DSO) .....	519
Configurar Apache para PHP .....	520
Configurar PHP utilizando php.ini .....	521
Directivas PHP en httpd.conf .....	521
php_admin_flag .....	521
php_admin_value .....	521
php_flag .....	522
php_value .....	522
Directivas PHP en php.ini .....	522
auto_append_file .....	522
auto_prepend_file .....	522
default_charset .....	523
disable_functions .....	523
display_errors .....	523
enable_dl .....	523
error_append_string .....	524
error_log .....	524
error-prepend_string .....	524
error_reporting .....	524
extension .....	525

extension_dir .....	526
implicit_flush .....	526
include_path .....	526
log_errors .....	527
magic_quotes_gpc .....	527
magic_quotes_runtime .....	527
max_execution_time .....	527
memory_limit .....	527
output_buffering .....	528
safe_mode .....	528
safe_mode_allowed_env_vars .....	528
safe_mode_protected_env_vars .....	529
track_errors .....	529
upload_max_filesize .....	529
upload_tmp_dir .....	529
<b>Trabajar con PHP .....</b>	<b>530</b>
Crear un script PHP sencillo desde la línea de comandos .....	530
Crear páginas Web PHP .....	530
Utilizar un script PHP como un Server-Side Include .....	531
Utilizar una página PHP para un directorio index .....	532
Utilizar archivos include .....	533
Mejorar el manejo de errores con PHP .....	535
Procesar formularios Web con PHP .....	535
Crear sesiones con PHP .....	538
Utilizar cookies HTTP para crear sesiones de usuario .....	538
Utilizar codificación de URL para crear sesiones de usuario .....	540
Finalizar una sesión de usuario .....	541
Utilizar MySQL con PHP .....	542
Crear una página PHP sencilla para acceder a la base de datos MySQL .....	542
Asegurar archivos include PHP .....	545
Autentificación de usuarios con PHP y MySQL .....	546
<b>16. Utilizar Perl con Apache .....</b>	<b>551</b>
En este capítulo .....	551
Compilar e instalar mod_perl .....	552
Ejecutar scripts CGI utilizando mod_perl .....	553
No realice más trabajo del necesario .....	554
Crear un módulo mod_perl utilizando el API de Perl para Apache .....	555
Utilizar CGI.pm para escribir módulos mod_perl .....	560
Precargar módulos Perl para ahorrar memoria .....	561
Seguir la pista de los módulos mod_perl en la memoria .....	562
Implementar ASP utilizando el módulo Apache::ASP .....	563

<b>17. Ejecutar servlets de Java y páginas JSP con Tomcat .....</b>	<b>567</b>
En este capítulo .....	567
Utilizar servlets .....	568
Instalar Tomcat .....	569
Instalar el último JDK para Tomcat.....	569
Instalar Tomcat y el módulo mod_jk .....	571
Configurar Tomcat .....	572
Configurar Tomcat para Apache .....	572
Configurar Tomcat para utilizar el Java Security Manager .....	577
Configurar Apache para Servlets y JSP .....	579
Trabajar con Tomcat .....	583
Desactivar el servicio HTTP por defecto de Tomcat .....	583
Iniciar y parar Tomcat .....	584
Iniciar Tomcat con un empaquetador de scripts de shell .....	584
Ejecutar servlets de Java .....	585
Ejecutar un ejemplo de servlets de Java mediante Tomcat .....	585
Ejecutar sus propios servlets o JSP .....	590
<b>Parte IV. Asegurar su sitio Web .....</b>	<b>593</b>
<b>18. Seguridad Web .....</b>	<b>595</b>
En este capítulo .....	595
Entender el concepto de seguridad Web .....	595
Los puntos de control .....	596
Punto de control 1: su red .....	598
Punto de control 2: el sistema operativo .....	599
Punto de control 3: software del servidor Web.....	600
Elegir una configuración segura .....	601
Consideraciones de política de seguridad .....	601
Una configuración de seguridad práctica para Apache .....	603
Utilizar un usuario y un grupo especializado para Apache .....	603
Utilice una estructura de directorios segura .....	604
Permisos de archivos y directorios apropiados .....	605
Archivo index del directorio .....	607
Desactivar el acceso por defecto .....	609
Desactivar invalidación de usuarios .....	610
La configuración "El cordero del sacrificio" .....	610
La configuración paranoica .....	611
Proteger su contenido Web .....	613
Guías de publicación de contenido .....	613
Proteger su contenido de robots.....	614
Excluir todos los robots.....	615
Permitir acceso completo a todos los robots .....	616

Excluir un solo robot .....	616
Activar un solo robot .....	616
Desactivar un solo archivo .....	616
<b>Registro y seguridad .....</b>	<b>616</b>
CustomLog y ErrorLog .....	617
Qué hacer si observa un acceso inusual en sus archivos de registro .....	618
<b>Asegurar su implementación CGI .....</b>	<b>619</b>
Evadir los riesgos CGI con un programa inteligente .....	619
Filtración de información .....	620
Consumo de los recursos del sistema .....	620
Burlarse de los comandos del sistema mediante scripts CGI .....	620
Las entradas del usuario realizan determinadas llamadas inseguras al sistema .....	621
El usuario puede modificar datos ocultos en páginas HTML .....	623
Entradas del usuario seguras .....	630
Empaquetar scripts CGI .....	634
suEXEC .....	634
CGIWrap .....	638
Ocultar pistas sobre sus scripts CGI .....	639
Utilice un alias de script que no sea estándar .....	640
Utilice nombres sin extensión para sus scripts CGI .....	640
Utilizar escáneres CGI .....	640
cgichk.pl .....	641
Whisker .....	643
Reducir riesgos SSI .....	644
<b>19. Asegurar Apache con SSL .....</b>	<b>647</b>
En este capítulo .....	647
Introducción a SSL .....	648
Cómo funciona SSL .....	649
Entender la encriptación .....	649
Entender los certificados .....	651
Transacciones basadas en certificados .....	652
Definir una Autoridad certificadora .....	654
Establecer SSL para Apache .....	655
Opciones SSL .....	656
Establecer OpenSSL .....	656
Requisitos previos de OpenSSL .....	656
Obtener OpenSSL .....	656
Compilar e instalar OpenSSL .....	657
Elegir el módulo mod_ssl para soporte SSL .....	660
Compilar e instalar mod_ssl .....	660
Configurar Apache para SSL basado en mod_ssl .....	661

Elegir Apache-SSL en lugar de mod_ssl para soporte SSL .....	664
Compilar e instalar parches Apache-SSL para Apache .....	664
Crear un certificado para el servidor Apache-SSL .....	665
Configurar Apache con Apache-SSL .....	666
Probar su conexión SSL .....	667
Obtener un certificado .....	668
Obtener un certificado para el servidor desde una CA comercial .....	668
Generar una clave privada .....	668
Generar un CSR .....	669
Crear una autoridad de certificación privada.....	670
Acceder a páginas SSL .....	671
<b>Parte V. Ejecutar Apache en Win32 .....</b>	<b>673</b>
<b>20. Instalar y ejecutar Apache para Windows .....</b>	<b>675</b>
En este capítulo .....	675
Requisitos del sistema .....	676
Cargar Apache para Windows .....	676
Instalar binarios de Apache .....	677
Ejecutar Apache .....	681
Ejecutar Apache automáticamente como un servicio Windows.....	681
Ejecutar Apache desde el menú Start .....	684
Gestionar Apache desde la línea de comandos.....	685
Ejecutar varios servicios Apache .....	685
<b>21. Configurar Apache para Windows .....</b>	<b>689</b>
En este capítulo .....	689
Sintaxis httpd.conf en Windows .....	690
Ajustar Apache para su funcionamiento .....	690
Probar la configuración de Apache .....	691
Gestionar Apache con Comanche .....	691
Configurar Apache para contenido dinámico .....	694
Ejecutar scripts CGI basados en Perl.....	695
Ejecutar scripts mod_perl .....	695
Ejecutar scripts PHP .....	697
Ejecutar extensiones ISAPI con mod_isapi .....	698
ISAPIReadAheadBuffer .....	698
ISAPILogNotSupported .....	699
ISAPIAppendLogToErrors.....	699
ISAPIAppendLogToQuery .....	699
UserDir en Windows .....	699

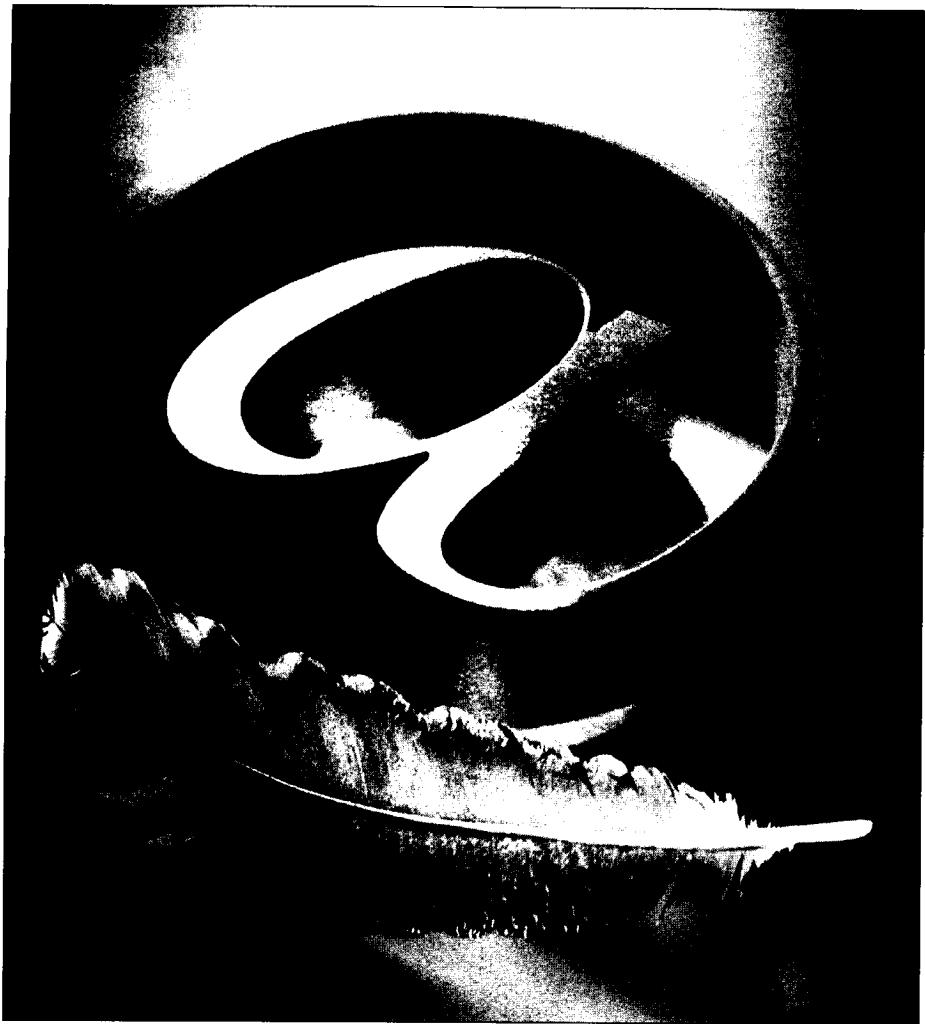
<b>Parte VI. Mejorar la escalabilidad .....</b>	<b>701</b>
<b>22. Apurando Apache .....</b>	<b>703</b>
En este capítulo .....	703
Utilizar hardware de alto rendimiento .....	704
CPU .....	704
RAM .....	704
Disco duro .....	705
Entender los acrónimos .....	707
Trucos en la elección de un disco duro .....	709
Poner a punto sus discos duros EIDE/IDE (Electrónica de dispositivos integrados o Electrónica de unidades inteligentes) en Linux .....	709
Comprobar las opciones de su disco duro con hdparam .....	711
Poner a punto el modo multisector para su disco duro .....	712
Activar acceso directo a memoria (Direct Memory Access, DMA) en su disco duro .....	713
Tarjeta ethernet .....	714
Poner a punto el sistema de archivos ext2 de Linux .....	714
Cambiar el tamaño del bloque del sistema de archivos ext2 .....	715
Poner a punto el sistema de archivos ext2 con e2fsprogs .....	716
Instalar e2fsprogs .....	716
Poner a punto su sistema de archivos con tune2fs .....	717
Comprobar y reparar un sistema de archivos ext2 con e2fsck .....	719
Poner a punto su sistema operativo .....	719
Compilar e instalar un kernel personalizado .....	719
Ajustar su sistema para aplicaciones Web en demanda .....	720
Controlar el número máximo de manejadores de archivos abiertos .....	720
Convertir el software de su servidor Apache de competitividad global .....	721
Poner a punto su red .....	723
Utilizar fast Ethernet .....	724
Entender y controlar el flujo de tráfico de red .....	724
Equilibrio de carga utilizando el servidor DNS .....	727
Utilizar hardware de equilibrio de carga .....	727
Poner a punto la configuración de Apache .....	729
Minimizar las búsquedas DNS .....	729
Apurar el servicio de archivos estáticos .....	729
Reducir el manejo de I/O para entregar páginas estáticas rápidamente .....	729
Reducir las llamadas al sistema y los manejos I/O para los archivos simbólicos .....	730
Poner a punto su configuración utilizando ApacheBench .....	732
Utilizar el caching para aumentar la velocidad .....	734
Meter los archivos muy utilizados en la memoria caché con mod_fcache ..	734

Adquirir habilidad con el servidor proxy-caché Squid .....	736
Compilar e instalar el servidor proxy-caché Squid .....	736
Configurar Squid .....	737
Iniciar su Squid .....	738
Personalizar Squid para satisfacer sus necesidades .....	739
Utilizar mod_backhand para una estancia de servidores Web .....	742
Poner a punto aplicaciones Web .....	743
Apurar los scripts mod_perl .....	743
Precargar sus módulos mod_perl .....	744
Caching conexiones de bases de datos .....	744
Ejecutar aplicaciones mod_perl en un conjunto parcial de hijos	
Apache .....	747
Utilizar FastCGI en lugar de mod_perl .....	749
<b>23. Crear una red de alta disponibilidad .....</b>	<b>753</b>
En este capítulo .....	753
Características de una red de alto nivel .....	754
Aumentar la seguridad DNS .....	754
Equilibrio de carga en su red .....	755
Distribuir solicitudes HTTP con Round-Robin DNS .....	755
Distribuir solicitudes HTTP con equilibradores de carga basados en hardware .....	756
Gestionar almacenamiento Web .....	758
RAID, SAN o dispositivos de almacenamiento .....	759
RAID de hardware .....	759
Las redes de almacenamiento (SAN) .....	759
Dispositivos de almacenamiento .....	760
Poner a punto sus discos duros .....	760
Obtener hdparam .....	761
Estimar el rendimiento de su unidad de disco .....	761
Mejorar el rendimiento de su unidad de disco .....	763
Ajustar el sistema de archivos ext2 .....	765
Cambiar el tamaño del bloque del sistema de archivos ext2 .....	765
Instalar e2fsprogs para ajustar el sistema de archivos ext2 .....	766
Comprobar y reparar un sistema de archivos ext2 con e2fsck .....	769
Aumentar la seguridad con un sistema de archivos journaling para Linux .....	769
Compilar e instalar ReiserFS .....	771
Montar el sistema de archivos ReiserFS .....	772
Utilizar un benchmark para ReiserFS .....	772
Compartir espacio de disco con el servidor NFS .....	775
Establecer un servidor NFS .....	775
Aspectos de seguridad del servidor .....	777

Establecer un cliente NFS .....	778
Optimizar el tamaño del bloque de caracteres de lectura / escritura .....	779
Establecer la unidad de transmisión máxima apropiada .....	782
Ejecuta el número óptimo de demonios NFS .....	783
Monitorizar los fragmentos de paquetes .....	783
Replicar contenido entre servidores Web .....	784
Utilizar rdist para distribuir archivos .....	784
Crear un sistema de archivos basado en RAM .....	788
Activar un sistema de archivos basado en RAM .....	788
Utilizar el sistema de archivos basado en RAM .....	790
Crear una red back-end segura .....	792
Fortificar su red Web .....	793
Utilizar Tripwire para proteger el contenido Web .....	794
Obtener Tripwire .....	795
Compilar Tripwire .....	795
Configurar la política Tripwire .....	799
Crear la base de datos Tripwire .....	803
Proteger el propio Tripwire .....	803
Ejecutar Tripwire para detectar integridad en el modo interactivo .....	804
Ejecutar Tripwire para detectar integridad de forma automática .....	806
Actualizar la base de datos Tripwire .....	807
Obtener un informe tripwire por correo electrónico .....	807
Asegurar Apache utilizando el Intrusion Detection System (LIDS) de Linux .....	809
Parchear, compilar e instalar el kernel con LIDS .....	810
Compilar, instalar y configurar LIDS .....	813
Administrar LIDS .....	815
Proteger archivos y directorios .....	816
Proteger su sistema utilizando las capacidades de Linux gestionadas por LIDS .....	823
Responder a un intruso .....	826
<b>Parte VII. Apéndices .....</b>	<b>827</b>
<b>Apéndice A. Códigos de estado HTTP 1.1 .....</b>	<b>829</b>
Códigos de estado de información (100–199) .....	829
Éxito en la solicitud del cliente (200–299) .....	830
Redirección de solicitudes (300–399) .....	831
Solicitud del cliente incompleta (400–499) .....	831
Errores del servidor (500–599) .....	833
<b>Apéndice B. Entender las expresiones regulares .....</b>	<b>835</b>

<b>Apéndice C. Recursos Apache online .....</b>	<b>839</b>
Recursos gratuitos .....	839
Sitios Web .....	839
Grupos de noticias Usenet .....	840
Grupos de noticias relacionados con servidores Web .....	840
Grupos de noticias relacionados con lenguajes de autor .....	841
Grupos de noticias relacionados con navegadores Web .....	842
Grupos de noticias de anuncios .....	842
Otros grupos de noticias WWW .....	842
Grupos de noticias Perl .....	843
Listas de correo .....	843
Recursos comerciales .....	843
Otros recursos relacionados .....	844
<b>Apéndice D. Contenido del CD-ROM .....</b>	<b>847</b>
Distribución del servidor Apache .....	847
Scripts de ejemplo en formato de texto .....	848
MySQL .....	848
OpenSSL .....	848
PHP .....	848
Perl y módulos relacionados .....	849
Tomcat .....	849
Solucionar errores .....	849





# Introducción

---

Bienvenido a Apache Server 2.0. Seguramente habrá oído hablar de Apache Server. De hecho, más del 60 por 100 de los administradores de toda la Web utilizan Apache. Apache es la plataforma de servidores Web de código fuente abierto más poderosa del mundo.

Como desarrollador Web profesional, investigador y administrador, considero que Apache es la solución perfecta para la mayor parte de los sitios Web. Apache 2.0 es una profunda revisión del servidor Apache. El grupo Apache creó originalmente una primera versión de un servidor Web altamente configurable, el cual se hizo popular rápidamente; en la versión 2, el grupo Apache se ha concentrado en la escalabilidad, en la seguridad y en el rendimiento. Las principales revisiones de código se han llevado a cabo para crear una arquitectura Apache realmente escalable.

Hoy en día, Apache es considerada la plataforma Web más utilizada. Aumentan día a día el número de corporaciones que aceptan este maravilloso código fuente abierto en su infraestructura IT. Son muchas las grandes compañías, como IBM, que ofrecen Apache entre sus productos. El futuro de Apache parece muy prometedor. Tanto si usted es nuevo en la utilización de Apache como si se trata de un administrador profesional del mismo, ahora es el momento de comenzar con Apache 2.0. Este libro le ayudará a hacerlo.

# Cómo está organizado este libro

El libro se compone de siete partes. A continuación le presentamos una breve descripción de las mismas.

## **Parte I. Comenzando**

En esta primera parte introduciré el servidor Web número uno en el mundo y le guiaré a través del proceso de obtención y compilación de Apache. Le mostraré cómo preparar y ejecutar Apache con el menor número de cambios posibles con respecto a la configuración de los archivos por defecto para que pueda preparar y ejecutar Apache lo más rápido posible. Esta parte finaliza con referencias completas a las directivas principales de Apache y a los módulos estándar para que, de este modo, pueda estar preparado para las tareas de administración de Apache.

## **Parte II. Administrar sitios Web**

Esta parte se centra en la administración de las tareas habituales de administración en la Web como son la creación de sitios web virtuales, la autenticación y autorización de usuarios para las distintas tareas, la monitorización, el registro, el redireccionamiento y la reescritura y este tipo de tareas en general. Aprenderá a crear y a administrar sitios Web virtuales. Dominará varios métodos de autenticación, de autorización y de control de acceso de usuarios. Aprenderá a monitorizar servidores Web y a adaptar archivos de registro para análisis.

## **Parte III. Ejecutar aplicaciones Web**

Esta parte se centra en las distintas posibilidades que existen para servir contenidos dinámicos utilizando Apache. Esta parte cubre los conceptos de Common Gateway Interface (CGI), Server-Side Includes (SSI), FastCGI, PHP, mod\_perl, y servlets de Java. Aprenderá a utilizar estas tecnologías rápidamente.

## **Parte IV. Asegurar su sitio Web**

Cualquier ordenador en Internet puede ser objeto de abuso o de intentos de manejo indebido. Siempre es una buena idea ser prudente y tomar las medidas de precaución adecuadas. En esta parte, aprenderá a hacer sus sitios Web más seguros y resistentes al ataque de hackers. Además, se le introducirá en los riesgos potenciales de la ejecución de programas SSI y CGI y en cómo tomar medidas preventivas para evitar estos riesgos. También aprenderá a activar el servicio Secure Socket Layer (SSL) utilizando módulos Apache para posibilitar el comercio electrónico

## **Parte V. Ejecutar Apache en Win32**

El servidor Apache está comenzando a ser muy popular en la plataforma Windows (Win32); cada vez más personas están probando Apache en platafor-

mas Windows. Con Apache 2.0, el rendimiento del servidor Web en esta plataforma se está haciendo cada vez más prometedor. En esta parte, aprenderá a instalar y a configurar Apache en la plataforma Win32.

## **Parte VI. Mejorar la escalabilidad**

En esta parte, discutiré cómo podemos aumentar la velocidad de Apache ajustando el sistema del servidor Web y optimizando varias configuraciones de servidores Apache. El capítulo ofrece gran cantidad de información sobre cómo sacar partido de un hardware de alto rendimiento y cómo ajustar discos duros y sistemas de archivos en Linux, para aumentar el rendimiento del sistema. Además, cubre los aspectos de caching y de ajuste relacionados con aplicaciones Web basadas en Perl.

## **Parte VII. Apéndices**

En esta parte, podemos encontrar cuatro apéndices que contienen los códigos de estado HTTP 1.1, información sobre las expresiones regulares, recursos online y sobre el contenido del CD-ROM que acompaña al libro.

# **Convenios utilizados en este libro**

No tiene que aprender ningún nuevo convenio para leer este libro, sino simplemente recordar que, cuando le piden que introduzca un comando, tiene que presionar la tecla Enter o Intro tras escribir la instrucción en el prompt de comandos. Se utiliza una fuente *Courier* para indicar un segmento de código.

Además podrá encontrar los siguientes elementos:

**NOTA:** El ícono Nota indica que es necesaria una mayor explicación.

**TRUCO:** El ícono Truco le ofrece algún modo de ahorrar algo de tiempo y esfuerzo.

**ADVERTENCIA:** El ícono Aviso le previene de un peligro potencial.

**CD-ROM:** El ícono CD-ROM le informa sobre archivos, programas y otras facilidades que puede encontrar en el CD-ROM.



**Parte I**

# **Comenzar**



# 1 Apache: el servidor número uno

---

## En este capítulo

1. Entendemos por qué Apache es tan popular.
2. Estudiamos la historia de Apache.
3. Echamos un vistazo al conjunto de características.
4. Examinamos la arquitectura Apache.
5. Revisamos las opciones de licencia.

Bienvenido a Apache, el servidor número uno del mundo. Si está acariciando la idea de ejecutar Apache, se encuentra en el lugar adecuado. Este capítulo introduce el modo en que Apache ejecuta un servidor Web.

Más del 60 por 100 de los servidores Web del mundo utilizan Apache, de acuerdo con un eminente servidor Web de una compañía dedicada a encuestas llamada Netcraft ([www.netcraft.co.uk/Survey/](http://www.netcraft.co.uk/Survey/)). Esta compañía publica periódicamente las estadísticas de los servidores más utilizados. La tabla 1.1 muestra las estadísticas publicadas en el momento en el que se escribió este capítulo. Si quiere conocer a los miembros de Apache, puede visitar [www.apache.org/info/apache\\_users.html](http://www.apache.org/info/apache_users.html).

**Tabla 1.1.** Estadística de Netcraft que muestra los servidores más utilizados

Servidor	Nov 2001	Porcentaje	Dic 2001	Porcentaje
Apache	7750275	61.88	8588323	63.34
Microsoft IIS	3307207	26.40	3609428	26.62
IPlanet	431935	3.45	383078	2.83
Zeus	174052	3.45	172352	1.27

## Popularidad de Apache

Lo conseguido por Apache es simplemente asombroso. Quién hubiese dicho que ese servidor de código fuente abierto podría vencer a sus dos mayores competidores comerciales, Microsoft y Netscape, como plataforma de servidores Web. Cada persona tiene sus razones para justificar la popularidad de Apache. Aquí están las mías:

- **Apache es un servidor altamente configurable de diseño modular.** Es muy sencillo ampliar las capacidades del servidor Web Apache. Cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada. Esto significa que hay una gran cantidad de módulos Apache disponibles para su utilización.
- **Apache es una tecnología gratuita de código fuente abierto.** El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto.
- **Apache trabaja con gran cantidad de Perl, PHP y otros lenguajes de script.** Perl destaca en el mundo del script y Apache utiliza su parte del pastel de Perl tanto con soporte CGI como con soporte mod\_perl.
- **Apache funciona en Linux y en otros sistemas de Unix.** Linux, acostumbrado a ser un sistema de operación desvalido, se encuentra ahora en los ruedos de las empresas de ordenadores. Linux y Apache van de la mano en el mundo empresarial de hoy en día. Considero que la aceptación de Linux en el mundo de los negocios ha hecho sencilla la entrada de Apache en ese territorio. Sin embargo, hay personas que argumentarían que fue la fama de Apache la que hizo que Linux encontrase su camino en ese mundo de un modo más sencillo. De cualquier modo, Apache y Linux constituyen una poderosa combinación. Otros sistemas de Unix como FreeBSD y Solaris, y el nuevo Mac OS X juegan también un papel importante en la ampliación de las perspectivas de los usuarios de Apache.

- **Apache también funciona en Windows.** Aunque Apache va a funcionar mucho mejor en Windows con la versión 2.0, ya se encontraba en el mercado de Windows con la versión 1.3.x. Veremos multitud de sistemas Windows conectados a Apache en vez de a Microsoft Internet Information Server (IIS) porque la arquitectura de Apache 2.0 le da el poder que necesita para competir.

## Apache: el comienzo

A continuación tenemos un poco de historia de Apache. Al principio, el NCSA (National Center for Super Computing Applications) creó un servidor Web que se convirtió en el número uno en 1995. Sin embargo, el principal desarrollador de servidores Web del NCSA abandonó el NCSA casi en ese mismo momento y el proyecto del servidor empezó a bloquearse. Entretanto, la gente que estaba utilizando el servidor Web de NCSA, empezó a intercambiar sus propios paquetes para el servidor y pronto pensaron que era necesario un foro para gestionarlos. Nació el grupo Apache.

El grupo utilizaba el código del servidor Web de NCSA y dio nacimiento a un nuevo servidor Web llamado Apache. Originariamente derivado del código central del servidor Web de NCSA y de un manojo de paquetes, hoy en día el servidor Web Apache es el lenguaje de la comunidad de los servidores Web. En los siguientes tres años, adquirió el papel de servidor líder del mercado. La primera versión (0.6.2) de Apache que fue distribuida al público se estrenó en abril de 1995. La versión 1.0 se estrenó el 1 de diciembre de 1995. El grupo Apache se amplió y se convirtió en un grupo sin ánimo de lucro. El grupo trabaja exclusivamente vía Internet. Sin embargo, el desarrollo del servidor Apache no está limitado en ningún sentido por el grupo.

Cualquiera que tenga el conocimiento para participar en el desarrollo del servidor o de sus módulos componentes, es bienvenido para hacerlo, aunque el grupo es la autoridad que finalmente decide qué se incluye en la distribución estándar de lo que se conoce como el servidor Apache. Esto permite que, literalmente, miles de desarrolladores de todo el mundo aporten nuevas características, localicen fallos, puertos para nuevas plataformas, etc. Cuando se envía código nuevo al grupo Apache, los miembros del grupo investigan los detalles, realizan las pruebas y las revisiones del control de calidad. Si están satisfechos, el código se integra en la distribución de Apache.

## La lista de características de Apache

Una de las principales características que presenta Apache es que funciona en plataformas virtuales muy utilizadas. Al principio, Apache se utilizaba para ser

el primer servidor Web basado en Unix, pero esto ya no es verdad. Apache no sólo funciona en la mayoría (prácticamente en todas) las versiones de Unix sino que, además, funciona en Windows 2000/NT/9x y en muchos otros sistemas operativos de escritorio y de tipo servidor como son Amiga OS 3.x y OS/2.

Apache presenta muchas otras características, entre ellas un elaborado índice de directorios; un directorio de alias; negociación de contenidos; informe de errores HTTP configurable; ejecución SetUID de programas CGI; gestión de recursos para procesos hijos; integración de imágenes del lado del servidor; reescritura de las URL; comprobación de la ortografía de las URL; y manuales online.

El resto de características importantes de Apache son:

- **Soporte del último protocolo HTTP 1.1:** Apache es uno de los primeros servidores Web en integrar el protocolo HTTP 1.1. Es totalmente compatible con el nuevo estándar HTTP 1.1 y al mismo tiempo sigue siendo compatible con HTTP 1.0. Apache está preparado para todas las novedades del nuevo protocolo. Por ejemplo, antes de HTTP 1.1, un navegador Web tenía que esperar una respuesta del servidor Web antes de poder emitir otra petición. Con el surgimiento de HTTP 1.1, esto ha dejado de ser así. Un navegador Web puede enviar solicitudes en paralelo, las cuales ahorran ancho de banda dejando de trasmitir las cabeceras HTTP en cada solicitud. De algún modo estamos ofreciendo un estímulo del lado del usuario final porque los archivos solicitados en paralelo aparecerán antes en el navegador.
- **Sencillo, con la configuración basada en un poderoso archivo:** el servidor Apache no posee una interfaz de usuario gráfica para su administración. Se trata de un sencillo archivo de configuración llamado `httpd.conf` que se puede utilizar para configurar Apache. Únicamente necesita su editor de texto favorito. Sin embargo, es lo suficientemente flexible para permitirle repartir la configuración de su host virtual en múltiples archivos para no sobrecargar un único archivo `httpd.conf` con toda la gestión de las múltiples configuraciones de servidores virtuales.
- **Soporte para CGI (Common Gateway Interface):** Apache soporta CGI utilizando los módulos `mod_cgi` y `mod_cgid`. Es compatible con CGI y aporta características extendidas como personalización de las variables de entorno y soporte de reparación de errores o debugging, que son difíciles de encontrar en otros servidores Web. Ver capítulo 12 para obtener más detalles.
- **Soporte de FastCGI:** no todo el mundo escribe sus CGI en Perl, ¿cómo pueden hacer sus aplicaciones CGI más rápidas? Apache también tiene una solución para esto. Utilice el módulo `mod_fcg` para implementar un entorno FastCGI dentro de Apache y haga que sus aplicaciones FastCGI arranquen rápidamente. Ver el capítulo 14 para obtener los detalles.

- **Soporte de host virtuales:** Apache es además uno de los primeros servidores Web en soportar tanto host basados en IP como host virtuales. Ver el capítulo 6 para obtener los detalles.
- **Soporte de autentificación HTTP:** Apache soporta autentificación básica basada en la Web. Está también preparado para autentificación basada en la digestión de mensajes, que es algo que los navegadores Web populares ya han implementado. Apache puede implementar autentificación básica utilizando tanto archivos estándar de contraseña como los DBM, llamadas a SQL o llamadas a programas externos de autentificación. Ver el capítulo 7 para obtener los detalles.
- **Perl integrado:** Perl se ha convertido en el estándar para la programación de scripts CGI. Apache es seguramente uno de los factores que hacen de Perl un lenguaje de programación CGI tan popular. Apache se encuentra más cerca de Perl que nunca. Puede bajar un script CGI basado en Perl a la memoria utilizando su módulo `mod_perl`, y reutilizarlo tantas veces como necesite. Este proceso elimina las desventajas del arranque que se encuentran asociadas a menudo con los lenguajes de interpretación como Perl. Ver el capítulo 16 para obtener los detalles.
- **Soporte de scripts PHP:** este lenguaje de script ha comenzado a ser muy utilizado y Apache ofrece un amplio soporte de PHP utilizando el módulo `mod_php`. Ver el capítulo 15 para obtener los detalles.
- **Soporte de servlets de Java:** los servlets de Java y las Java Server Pages (JSP) se están convirtiendo en algo muy común en los sitios Web dinámicos. Puede ejecutar servlets de Java utilizando el premiado entorno Tomcat con Apache. Ver el capítulo 17 para obtener los detalles.
- **Servidor proxy integrado:** puede convertir Apache en un servidor proxy caché. Sin embargo, la implementación actual del módulo opcional de proxy no soporta HTTP proxy o el último protocolo HTTP 1.1. Se está planeando actualizar este módulo muy pronto. Ver el capítulo 10.
- **Estado del servidor y adaptación de registros:** Apache le da una gran cantidad de flexibilidad en el registro y la monitorización del estado del servidor. El estado del servidor puede monitorizarse mediante un navegador Web. Además, puede adaptar sus archivos de registro a su gusto. Ver el capítulo 8 para obtener los detalles.
- **Soporte de Server Side Includes (SSI):** Apache ofrece un conjunto de Server Side Includes que añaden una gran cantidad de flexibilidad para el desarrollador del sitio Web. Ver el capítulo 13 para obtener los detalles.
- **Soporte de Secured Socket Layer (SSL):** puede crear fácilmente un sitio Web SSL utilizando OpenSSL y el módulo `mod_ssl` de Apache. Ver el capítulo 19 para obtener los detalles.

# Entender la arquitectura de Apache 2.0

Apache Server 2.0 hace de Apache una solución Web más flexible, más transportable y más escalable que nunca. La nueva versión 2.0 ofrece muchas mejoras; las principales se discuten en las siguientes secciones.

## Módulos multiproceso

El primer cambio principal en Apache 2.0 es la introducción de los módulos multiproceso (MPM). Para entender por qué se han creado los módulos MPM, necesita entender cómo trabajaba antes Apache. La versión 1.3 de Apache y las anteriores utilizaban una arquitectura sin bifurcaciones. En este tipo de arquitecturas, un proceso padre de Apache se bifurca como un conjunto de procesos hijos, los cuales son los que en realidad sirven las solicitudes. El proceso padre simplemente monitoriza los hijos y produce o elimina procesos hijos basándose en la cantidad de solicitudes recibidas. Desgraciadamente, este modelo no trabajaba bien bajo plataformas que no estaban centradas en proceso como en el caso de Windows. De modo que el grupo Apache propuso una solución basada en los MPM.

Cada MPM es responsable de iniciar los procesos del servidor y de servir las solicitudes vía procesos hijos o hilos dependiendo de la implementación MPM. Hay disponibles muchas MPM. Se discutirán en las siguientes secciones.

### El MPM prefork

Este MPM sin bifurcaciones mimetiza al Apache 1.3 o a otras arquitecturas anteriores, creando un grupo de procesos hijos para servir las solicitudes. Cada proceso hijo tiene un solo hilo. Por ejemplo, si Apache inicia 30 procesos hijo, puede servir 30 solicitudes simultáneamente. Si algo va mal y el proceso hijo muere, únicamente se pierde una solicitud. El número de procesos hijo está controlado fijando un mínimo y un máximo. Cuando aumenta el número de solicitudes, un nuevo proceso hijo se añade hasta que se alcanza el máximo. Del mismo modo, cuando falla una solicitud, cualquier proceso hijo extra es eliminado.

### El MPM threaded

Este MPM permite soporte de hilos en Apache 2.0. Es como el MPM prefork, pero en vez de que cada proceso hijo tenga un solo hilo, cada proceso hijo puede tener un número determinado de hilos. Cada hilo dentro de un proceso hijo puede servir una solicitud distinta. Si Apache inicia 30 procesos hijo en los que cada hijo puede tener un máximo de 10 hilos, entonces Apache podrá servir  $30 \times 10 = 300$  solicitudes simultáneamente.

Si algo va mal con un hilo, por ejemplo, un módulo experimental hace que el hilo muera, entonces el proceso entero muere. Esto significa que se perderán

todas las solicitudes que han servido los hilos dentro del proceso hijo. Sin embargo, como las solicitudes están distribuidas en procesos hijo separados, la muerte de un proceso hijo afecta como máximo a  $1/n$  del total de las conexiones, donde  $n$  representa el número de conexiones simultáneas.

Los procesos se añaden o se eliminan monitorizando su conteo de hilos de repuesto. Por ejemplo, si un proceso tiene menos de un número mínimo de hilos de repuesto, se añade un nuevo proceso. Del mismo modo, cuando un proceso tiene un máximo de número de hilos parados, se elimina.

Todos los procesos funcionan con el mismo usuario e ID de grupo asignados por el servidor Apache.

Dado que los hilos son más eficaces en cuanto a recursos que los procesos, este MPM es muy escalable.

## **El MPM perchild**

Esto también es nuevo en Apache 2.0. En este modelo MPM, un número de procesos hijo se inicia con un número determinado de hilos. Según aumenta la carga de solicitudes, el proceso va añadiendo nuevos hilos a medida que los necesita. Cuando el conteo de solicitudes se reduce, los procesos disminuyen sus conteos de hilos utilizando un conteo mínimo y máximo fijos.

La diferencia clave entre este módulo y el MPM threaded es que el proceso de conteo es estático y además cada proceso puede ejecutarse utilizando un usuario y un ID de grupo distintos. Esto facilita la ejecución de distintos sitios Web virtuales bajo distintos usuarios e ID de grupos. Ver el capítulo 6 para obtener los detalles.

## **El MPM winnt**

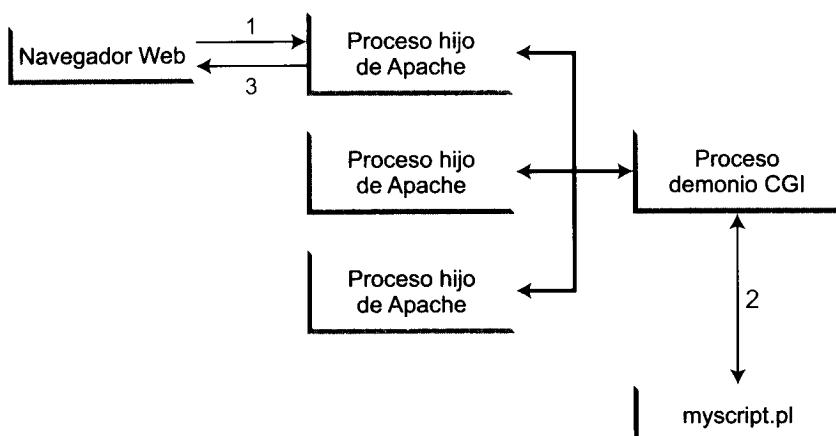
Este es el MPM para la plataforma Windows, incluido Windows 2000, Windows NT y Windows 9x. Se trata de un módulo multihilo. Utilizando este módulo, Apache creará un proceso padre y un proceso hijo. El proceso hijo crea todos los hilos que sirve la solicitud. Además, este módulo saca partido de algunas llamadas a funciones de Windows, que le permiten funcionar mejor en la plataforma Windows que las versiones anteriores del servidor Apache.

## **Filtrado I/O**

Ahora Apache 2.0 proporciona arquitectura para I/O jerarquizada. Esto significa que un output de un módulo puede convertirse en un input de otro módulo. El efecto de este filtrado es muy interesante. Por ejemplo, el output producido por scripts de CGI, que es procesado por el módulo `mod_cgi`, puede ahora pasarse al módulo `mod_include` responsable de las SSI. En otras palabras, los scripts de CGI pueden producir un output en forma de etiquetas SSI, que se pueden procesar antes de que el output final se envíe al navegador Web. Estarán disponibles en el futuro muchas otras aplicaciones de filtrado O/I.

# El nuevo demonio CGI

Dado que muchos de los módulos MPM utilizan hilos, ejecutar scripts CGI se convierte en algo engorroso cuando un hilo transfiere una solicitud. El módulo `mod_cgi` sigue funcionando, pero no de forma óptima para las MPM threaded. Por ese motivo se añadió `mod_cgid`. El módulo `mod_cgid` crea un proceso demonio, que produce procesos CGI e interacciona con hilos de forma más eficaz. La figura 1.1 muestra cómo se sirve un CGI solicitado por un script llamado `myscript.pl`.



**Figura 1.1.** Cómo trabaja el demonio CGI con un proceso hijo de Apache

A continuación tenemos el modo en que se ejecutan los scripts CGI:

1. Cuando la solicitud CGI llega a un hilo dentro de un proceso hijo, pasa la respuesta al demonio CGI.
2. El demonio CGI produce el script CGI y pasa los datos generados por el script CGI al hilo del proceso hijo.
3. El hilo devuelve el dato al navegador Web.

Cuando el servidor principal de Apache se inicia, también se inicia el demonio CGI y establece una conexión socket. Por lo tanto, cuando se crea un nuevo proceso hijo, hereda la conexión socket y por tanto no tiene necesidad de crear una conexión con el demonio CGI para cada solicitud. El proceso completo favorece la ejecución en el entorno con hilos.

## Apache es portable en tiempo de ejecución

Para desarrollar la visión del grupo Apache que consistía en la creación del servidor Web más popular en el mundo, se hizo patente la necesidad de la

portabilidad de Apache en Apache 2.2. Antes de la versión que se encuentra actualmente en el mercado, Apache tenía portabilidad interna, lo que hacía el código base menos manejable. De modo que el grupo Apache introdujo el Apache Portable Runtime (APR). El propósito de APR es proporcionar una sencilla interfaz de C a funciones específicas de plataforma para que se pueda escribir el código una sola vez.

Esto permite que Apache funcione mejor en plataformas como Apache Windows, BeOS, Amiga y OS/2. Ya que gracias a APR muchos programas como ApacheBench, pueden ejecutarse en estas plataformas.

## Entender la licencia de Apache

Los software gratuitos como Apache, Perl (Practical Extraction and Reporting Language) y Linux (un sistema operativo clon de Unix basado en x86) han de soportar una gran presión debido a la decisión de Netscape de fabricar el Netscape Comunicator, uno de los navegadores Web más populares, disponible de forma gratuita con su proyecto Mozilla. Desgraciadamente, el software gratuito como Apache, Perl y Linux no comparten los mismos acuerdos de licencia y se ha creado confusión asociando estos paquetes en la misma categoría de licencia.

Todo software gratuito trata de serlo para todos. Sin embargo, hay algunas restricciones legales que fuerzan las licencias de software individuales. Por ejemplo Linux, que es gratuito desde GNU Public License (GPL), exige que cualquier cambio en Linux se haga público. Apache, por su lado, no necesita que los cambios hechos en Apache se hagan públicos.

En pocas palabras, piense en Apache como un software gratuito, un software con derechos de autor publicado por el Grupo Apache. No es ni de dominio público ni de prueba. Además no olvide que Apache no está cubierto por GPL. El documento original de la licencia del software Apache se puede ver a continuación en el listado 1.1. En el listado 1.2 puede ver una versión traducida del mismo.

**Listado 1.1.** Licencia original del software Apache

```
/* =====
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 2000-2001 The Apache Software Foundation. All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
```

\* the documentation and/or other materials provided with the  
\* distribution.  
\*  
\* 3. The end-user documentation included with the redistribution,  
\* if any, must include the following acknowledgment:  
\* "This product includes software developed by the  
\* Apache Software Foundation (<http://www.apache.org/>)."  
\* Alternately, this acknowledgment may appear in the software itself,  
\* if and wherever such third-party acknowledgments normally appear.  
\*  
\* 4. The names "Apache" and "Apache Software Foundation" must  
\* not be used to endorse or promote products derived from this  
\* software without prior written permission. For written  
\* permission, please contact apache@apache.org.  
\*  
\* 5. Products derived from this software may not be called "Apache",  
\* nor may "Apache" appear in their name, without prior written  
\* permission of the Apache Software Foundation.  
\*  
\* THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED  
\* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES  
\* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE  
\* DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR  
\* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,  
\* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT  
\* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF  
\* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND  
\* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,  
\* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT  
\* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
\* SUCH DAMAGE.  
\* ======  
\*  
\* This software consists of voluntary contributions made by many  
\* individuals on behalf of the Apache Software Foundation. For more  
\* information on the Apache Software Foundation, please see  
\* <<http://www.apache.org/>>.  
\*  
\* Portions of this software are based upon public domain software  
\* originally written at the National Center for Supercomputing  
Applications,  
\* University of Illinois, Urbana-Champaign.  
\*/

### Listado 1.2. Licencia traducida del software Apache

/\* ======  
\* La licencia del software Apache, Versión 1.1  
\*  
\* Copyright (c) 2000-2001 The Apache Software Foundation. Todos los  
\* derechos reservados.  
\*  
\* La redistribución y uso de la fuente y los binarios, con o sin  
\* modificaciones, está permitida bajo las siguientes condiciones:  
\*  
\* 1. La redistribución del código fuente debe reproducir este mismo  
\* copyright y estar bajo las condiciones que este implica.  
\*  
\* 2. La redistribución en formato binario debe reproducir este mismo

\* copyright y estar bajo las condiciones que este implica en la  
\* documentación y/o con los otros materiales provistos con la  
\* distribución.  
\*  
\* 3. Todo aquello que utilice las características o partes de este  
\* software debe mostrar la siguiente sentencia:  
\* "Este producto incluye software desarrollado por el grupo Apache  
\* para su utilización en el proyecto del servidor HTTP Apache  
\* (<http://www.apache.org/>)."  
\* Esta sentencia puede aparecer también en el software en sí,  
\* siempre y cuando esa sentencia aparezca normalmente.  
\*  
\* 4. Los nombres "Apache" y "Apache Software Foundation" no se pueden  
\* utilizar para promocionar productos derivados de este software  
\* sin permiso escrito. Para conseguir este permiso escrito,  
\* contactar con apache@apache.org.  
\*  
\* 5. Los productos derivados de este software no podrán llamarse  
\* "Apache", y no podrá aparecer "Apache" en el nombre del producto  
\* sin permiso expreso del grupo Apache.  
\*  
\* EL GRUPO APACHE PROPORCIONA ESTE SOFTWARE 'TAL CUAL' INCLUSO SIN  
\* ASUMIR NINGUNA GARANTÍA, PERO NO SOLAMENTE, LA GARANTÍA  
\* MERCANTIL IMPLÍCITA O DE CONVENIENCIA PARA UN PROPÓSITO  
\* PARTICULAR. EN NINGÚN CASO DEBERÁN SER RESPONSABILIZADOS NI EL  
\* GRUPO APACHE NI NINGUNO DE SUS COLABORADORES POR DAÑOS O  
\* PERJUICIOS (INCLUSO, PERO NO LIMITADO A, BÚSQUEDA DE BIENES O  
\* SERVICIOS SUBSTITUTOS; PÉRDIDA DE DATOS, USO O LUCRO; O  
\* INTERRUPCIÓN LABORAL.) EXISTE ADVERTENCIA DE LA POSIBILIDAD DE  
\* ESTOS DAÑOS, BIEN SEAN CAUSADOS DE FORMA DIRECTA, INDIRECTA,  
\* ACCIDENTAL, ESPECIAL, EJEMPLAR O CONSECUENTE, CUALESQUIERA QUE SEAN  
\* SUS CAUSAS Y EN CUALQUIER ESQUEMA DE RESPONSABILIDAD LEGAL, SEA POR  
\* CONTRATO DE FORMA ESTRICTA, O POR OMISIÓN (INCLUYENDO NEGLIGENCIA  
\* Y OTRAS.)  
\* ======  
\*  
\* Este software consiste en contribuciones voluntarias hechas por  
\* muchas personas pertenecientes al Grupo Apache. Para más información  
\* sobre Apache Software Foundation, por favor visite la Web  
\* <<http://www.apache.org/>>.  
\*  
\* Parte de este software está basado en software de dominio público  
\* escrito originalmente por el Centro Nacional de Aplicaciones de  
\* Superconmutación, de la Universidad de Illinois, Urbana-Champaign.  
\*/



# 2 Obtener e instalar Apache

---

## En este capítulo

1. Encontramos la última fuente de Apache.
2. Analizamos las necesidades del sistema para Apache.
3. Bajamos el software.
4. Compilamos la fuente.
5. Instalamos binarios en su sistema.
6. Mantenemos al día el desarrollo de Apache.

Aunque compilar los propios binarios de Apache puede parecer mucho trabajo, merece la pena el esfuerzo. A medida que nos familiarizamos con Apache, aprendemos que es el único servidor Web que ofrece prácticamente todas (si no más) las funcionalidades de un servidor comercial a gran escala, al tiempo que le permite ver cómo están implementadas estas funcionalidades en la fuente. Encuentro este aspecto de Apache fascinante.

Para las personas que no son programadoras de C pero que siguen necesitando un servidor Web poderoso y gratuito, sin embargo, juguetear con esa cantidad de

código ANSI C no es precisamente un pasatiempo agradable. Afortunadamente, no hay por qué preocuparse, Apache se encuentra tanto en código fuente como en paquetes binarios preconstruidos. Este capítulo discute la instalación de Apache desde el código fuente y desde los binarios preconstruidos.

## La fuente oficial de Apache

Independientemente del sitio del que obtenga el software gratuito (código fuente o archivos binarios) de Internet, asegúrese de que no lo está obteniendo de un sitio Web o FTP desconocido. Para explicar lo que quiero decir con desconocido pondré un ejemplo. Imagine que quiere obtener software de un navegador Web basado en Java desarrollado por Sun Microsystems. Necesita ser capaz de hacer una serie de verificaciones de autenticidad, lo que suele ser bastante complicado en Internet; por lo que tendrá que acceder a sitios muy visitados. Por ejemplo, no debería obtenerlo de un sitio FTP con un hostname del tipo dialup-666.someforeignisp.net.ch; debería buscarlo probablemente en algún sitio de java.sun.com, ya que no puede estar seguro de que 666.someforeignisp.net.ch tenga completamente verificado el software para cualquier daño oculto. Creo que puede darse cuenta de lo que quiero decir.

Por suerte para nosotros, los desarrolladores y los que mantienen Apache nos aseguran un sitio oficial para obtener software de Apache. El sitio Web oficial de Apache es [www.apache.org](http://www.apache.org). Este sitio contiene la última versión estable de Apache, la que está en el mercado de Apache, parches, módulos Apache, etc. Aquí es donde usted debería acudir para todas las necesidades Apache, aunque debería redirigirse a un sitio mirror que se encuentre geográficamente cerca de usted para ayudarle a reducir el ancho de banda y la congestión de red. Además puede utilizar las firmas Pretty Good Privacy (PGP) aportadas por el sitio para verificar la autenticidad del código fuente. Si no sabe utilizar las PGP, diríjase a la siguiente dirección Web [www.pgp.net](http://www.pgp.net).

**NOTA:** *Usuarios de Windows, por favor lea el capítulo 20 para obtener los detalles específicos de su plataforma. Este capítulo está dedicado principalmente a la instalación Linux y la utilización de Linux como ejemplo de plataforma. Las instrucciones de este capítulo se aplican a la mayoría de los sistemas Unix.*

## Requisitos del sistema

Apache se ejecuta en prácticamente cualquiera de las plataformas que se utilizan hoy en día, incluido; FreeBSD; OpenBSD; NetBSD; BSDI; Amiga OS 3.x;

Mac OS X; SunOS; Solaris; IRIX; HPUX; Digital Unix; UnixWare; AIX; SCO; ReliantUNIX; DGUX; OpenStep/Mach; DYNIX/ptx; BeOS y Windows.

## Requisitos para construir Apache desde la distribución de la fuente

Si está pensando en construir Apache desde la fuente, que es lo que realmente recomiendo ya que si lo compila por sí solo le dará un servidor muy mediocre, entonces necesita asegurarse de que su sistema cumple los requisitos de la tabla 2.1.

**NOTA: Estos requisitos son para construir Apache desde el código fuente únicamente. Para ejecutar Apache en su sistema, navegue por los requisitos de la siguiente sección.**

Tabla 2.1. Requisitos para construir Apache desde la fuente

Recurso	Necesidad	Requisitos
Espacio en el disco	Obligatorio	Se necesitan aproximadamente 12MB de espacio en el disco para compilar e instalar Apache desde la distribución de la fuente. Sin embargo, si añade muchos módulos que no formen parte de la distribución estándar de la fuente, entonces aumenta la necesidad de espacio. Una vez que está instalado Apache, únicamente necesita unos 5 MB de espacio en el disco. Le recomiendo que mantenga el código fuente en su disco duro hasta que termine de leer este libro.
Compilador ANSI C	Obligatorio	Es necesario tener el compilador de ANSI C. Se recomienda el compilador GNU C (GCC) de la Free Software Foundation para la mayor parte de los sistemas. Debería tener la versión 2.7.2 o una superior. Puede encontrar el GCC en <a href="http://www.gnu.org">www.gnu.org</a> . La mayoría de los sistemas Linux, como Red Hat Linux, traen consigo la última versión estable del compilador GCC.

Recurso	Necesidad	Requisitos
Intérprete de Perl 5	Recomendado	No necesita Perl para compilar Apache, pero parte de los scripts de soporte, como apxs, split-logfile, log_server_status y dbmmanage, que se encuentran en el directorio de soporte de su distribución de fuente, son scripts de Perl. Únicamente necesita Perl si pretende utilizar estos scripts. Realmente le recomiendo que instale Perl en su sistema. La versión 5.003 o una versión superior funcionarán perfectamente.
Soporte de Dynamic Shared Object (DSO)	Opcional	En lugar de compilar módulos en el binario de Apache, puede crear módulos dinámicos llamados DSO, que se pueden bajar mediante el archivo httpd.conf al arrancar. Los módulos DSO le permiten realizar experimentos con los módulos y las configuraciones con mayor libertad que compilando todo en httpd. Actualmente, hay disponible soporte de DSO para Linux, FreeBSD; OpenBSD; NetBSD; BSDI; SunOS; Solaris; IRIX; HPUX, Digital Unix, UnixWare, AIX, SCO, ReliantUNIX, DGUX, Darwin/Mac OS, OpenStep/Mach y DYNIX/ptx. Tenga en cuenta que mientras que un servidor Apache está utilizando un módulo DSO será aproximadamente un 5 por 100 más lento durante el tiempo de ejecución. Además, DSO no está disponible en todas las plataformas. Por estos motivos, no recomiendo DSO en un entorno de producción. Sin embargo, es fantástico experimentar con módulos en desarrollo o en un entorno de prueba.

## Requisitos para ejecutar un servidor Web Apache

Antes de instalar Apache en su servidor Web, debe asegurarse de que su servidor Web tiene el "poder" suficiente para ejecutarlo.

Afortunadamente, Apache no necesita grandes recursos en el ordenador para funcionar. Funciona perfectamente en un sistema Linux de 5 a 12MB de espacio en el disco duro y 8MB de RAM. Sin embargo, no sólo querrá instalar Apache sino que, además, querrá ejecutar Apache para servir páginas Web, lanzar procesos CGI y sacar partido de todo el maravilloso material que la Web tiene que ofrecer. En ese caso, necesitará espacio en el disco y en la RAM suficientes para satisfacer sus necesidades. Puede llevar esto a cabo de dos modos: pidiéndole a alguien que ejecute un sitio similar con Apache y determinando qué tipo de recursos del sistema se están utilizando o puede tratar de entender sus necesidades reales una vez que tenga instalado Apache en su sistema.

En el último caso, puede utilizar ciertas utilidades del sistema como son `ps`, `top`, para conocer la cantidad de memoria que se está utilizando en un proceso Apache determinado. Puede determinar entonces la cantidad total de memoria necesaria multiplicando la cantidad de memoria que se utiliza en un solo proceso por el número total de procesos Apache que se ejecutarán en el momento álgido (ver la directiva `MaxSpareServers` en el capítulo 4). Esto le dará una estimación adecuada de las necesidades de RAM de su sitio con Apache. Si está pensando en ejecutar varios programas CGI en su servidor Apache, tiene que determinar también la cantidad de memoria necesaria para estos programas, y tener en cuenta esta necesidad añadida. Una forma de determinar las necesidades de memoria para los programas CGI es ejecutar el programa CGI y utilizar la utilidad `top` para ver cuánta memoria utiliza y de ese modo multiplicar esa cantidad de memoria por el número de solicitudes CGI que necesita para ser capaz de cumplirlas simultáneamente.

Los requisitos del disco para el código fuente de Apache o para los archivos binarios no deberían constituir una preocupación para casi ninguno de los sistemas porque los binarios de Apache no necesitan más de 1MB de espacio y el archivo fuente ocupa unos 5MB. Debe prestar atención, sin embargo, a los archivos de registro que Apache crea, porque cada entrada de registro ocupa más de 80 bytes de espacio en el disco. Si espera obtener alrededor de 100,000 entradas en un día, por ejemplo, su archivo de registro de acceso deberá ser de 8,000,000 bytes. En el capítulo 8, aprenderá cómo alternar los archivos de registro vaciando o archivando el archivo de registro y reemplazándolo por uno nuevo.

Para finalizar, determine si tiene el suficiente ancho de banda para ejecutar un servidor Web. Estimar el requisito del ancho de banda no es tan sencillo pero puede calcular una cifra aproximada con un poco de matemáticas. Esto es lo que necesita:

- **La media del tamaño de las páginas Web de su sitio Web:** si aún no lo sabe, puede ejecutar el siguiente comando en el directorio de la ruta de directorios para calcular el tamaño medio de sus páginas web:

```
find path_to_doc_root -type f -name "*.html" -ls | \
awk 'BEGIN{ FILECNT = 0; T_SIZE = 0; } \
{ T_SIZE += $7; FILECNT++ } \'
```

```
END{print "Total Files:", FILECNT, \
"Total Size:", T_SIZE, \
"Average Size:", T_SIZE / FILECNT;}'
```

**NOTA:** No olvide reemplazar path/to/doc\_root por el verdadero directorio root de documentos de su sitio Web. Por ejemplo, para un sitio Web con una raíz de documentos /www/mysite/htdocs, el script anterior devolvería la siguiente salida:

```
Total Files: 332 Total Size: 5409725 Average Size: 16294.4
```

- **Número de páginas Web de tamaño medio que puede servir (asumiendo que Apache no tiene cuellos de botella e ignorando totalmente el ancho de banda utilizado por las solicitudes entrantes):** por ejemplo, imagine que tiene una conexión RDSI (128Kbits/sec) de Internet y su tamaño de archivo medio es 16K. Como 128 kilobits por segundo = 128/8 kilobytes por segundo = 16 kilobytes por segundo, puede enviar un archivo de tamaño medio por segundo. Si aumenta el número de solicitudes, con ese ancho de banda, probablemente no pueda servir una solicitud por minuto. En este caso, la sobrecarga de red se convierte en un cuello de botella si quiere permitir que N (donde N>1) usuarios simultáneos se conecten con su sitio Web. Por ejemplo, si tiene una conexión RDSI y quiere servir 12 usuarios al mismo tiempo por segundo cuando el tamaño medio es 16 K, necesita 12 x RDSI (128K) conexiones, que es en realidad una conexión T-1 (1.53 Mbps).

## Bajar el software

Antes de bajar el software de Apache por primera vez, debe tener en cuenta una serie de cosas. Se pueden encontrar dos versiones disponibles de Apache: una es una versión oficial y la otra es la versión beta que tiene el ultimo código fuente y las últimas características. Si encuentra la versión 2.0.2 de Apache y una versión llamada 2.3b3, entonces la primera versión es la oficial y la segunda es la versión beta. La tercera versión beta 2.3b3 (las versiones 2.3b1 y 2.3b2 salieron antes) probablemente será una versión estable, pero no es recomendable utilizar una versión beta para producción en un servidor Web. Para bajar la versión que quiera, diríjase a <http://www.apache.org/dist/httpd/>.

**TRUCO:** Para encontrar el servidor mirror de Apache más cercano geográficamente, ejecute el buscador de scripts de Apache en [www.apache.org/dyn/closer.cgi](http://www.apache.org/dyn/closer.cgi).

Este es el directorio de distribución del software de Apache. Aquí, puede encontrar tanto la versión oficial como la beta del software en varios paquetes comprimidos. Por ejemplo:

```
httpd_2.0.4.tar.Z  
httpd_2.0.4.tar.gz  
httpd_2.0.4.zip  
httpd_2.3b3.tar.gz  
httpd_2.3b3_win32.exe
```

Estos son ejemplos de distintos tipos de formatos de compresión que se utilizan para distribuir código fuente. Tiene que elegir el formato de compresión que su sistema puede manejar (en otras palabras, asegúrese de que tiene la utilidad de descompresión de código). Normalmente con Linux, necesita tener las utilidades tar, gnuzip o gzip para descomprimir los archivos. Por ejemplo, para descomprimir el archivo httpd\_version.tar.gz (en el que version es cualquier versión que haya bajado como por ejemplo la 2.0.4) en un sistema Linux, utiliza el comando tar xvzf httpd\_version.tar.gz. Podría utilizar también el comando gzip -d httpd\_version.tar.gz; tar xvf httpd\_version.tar, que descomprimirá y extraerá todos los archivos en un subdirectorio mientras que mantiene intacta la ruta relativa para cada archivo.

Windows crea la auto extracción de archivos comprimidos para la versión de Apache. Cualquier archivo se puede extraer simplemente ejecutando el archivo que hemos bajado. Para la instalación específica en Windows y para saber los detalles de configuración, debe saltarse el resto del capítulo y dirigirse al capítulo 20.

Los binarios se guardan normalmente en un directorio distinto en el que cada sistema operativo tiene su propio subdirectorio. Tenga en cuenta que si su sistema operativo no aparece en el directorio de binarios, eso no significa necesariamente que no soporte ese sistema operativo. Lo único que significa es que no hay nadie del grupo de desarrollo de Apache o de los grupos colaboradores que tenga compilado un archivo binario para su sistema hasta ahora. Probablemente encontrará binarios para los sistemas Linux, FreeBSD, Solaris, NetBSD, OS2, AIX, Ultrix, HPUX y IRIX.

## Instalar Apache desde el código fuente

Instalar Apache compilando el código desde la distribución de la fuente es el mejor método de instalación porque permite configurar el servidor para satisfacer sus necesidades. Cualquier instalación binaria que baje tendrá alguna otra configuración que no podrá alterar para cubrir sus necesidades.

Por ejemplo, si baja e instala un binario que tenga soporte CGI, puede que tenga que mantener el soporte CGI aunque nunca vaya a ejecutar programas CGI.

Si el módulo CGI está configurado como un módulo dinámico compartido, entonces puede inutilizarlo fácilmente; sin embargo, si el soporte está construido de forma estática en el binario, entonces tendrá que dejarlo como está. Si compila un servidor Apache desde la fuente de distribución obtendrá los componentes que necesite sin gasto de procesos o de espacio en el disco.

**NOTA:** Baje la distribución de la fuente del sitio Apache oficial o del sitio mirror adecuado.

## Configurar la fuente de Apache

La distribución de fuente de Apache incluye un script llamado `configure` que le permite configurar el árbol fuente antes de compilar e instalar los binarios. Desde el directorio de distribución de fuente de Apache, puede ejecutar este script del siguiente modo:

```
./configure --prefix=apache_installation_dir
```

La opción `--prefix` le dice a Apache que instale los binarios, otras configuraciones necesarias y los archivos de soporte en `apache_installation_dir`. Por ejemplo:

```
./configure --prefix=/usr/local/apache
```

Aquí la fuente Apache se configurará de modo que todos los binarios y archivos de soporte se instalarán en el directorio `/usr/local/apache`.

Hay muchas opciones que puede utilizar con el script `configure`. La tabla 2.2 muestra todas las opciones de configuración disponibles.

**Tabla 2.2.** Las opciones del script de configuración

Opción	Significado
<code>--cache-file=file</code>	Los resultados de la prueba del caché se encuentran en <code>file</code> .
<code>--help</code>	Imprime este mensaje.
<code>--no-create</code>	No crear archivos de salida.
<code>--quiet or -silent</code>	No imprimir mensajes 'checking...'.
<code>--version</code>	Imprimir la versión de <code>autoconf</code> que creó el directorio y los nombres de archivo de configuración.
<code>--prefix=prefix</code>	Instala los archivos independientes de arquitectura en <code>[/usr/local/apache2]</code> .

Opción	Significado
--exec-prefix=eprefix	Instala los archivos dependientes de arquitectura en <i>eprefix</i> [same as prefix].
--bindir=dir	Ejecutables de usuario en dir [EPREFIX/bin].
--sbindir=dir	Ejecutables del administrador del sistema en dir [EPREFIX/sbin].
--libexecdir=dir	Ejecutables de programas en dir [eprefix/libexec].
--datadir=dir	Datos independientes de arquitectura de sólo lectura en dir[prefix/share].
--sysconfdir=dir	Datos de una máquina única de sólo lectura en dir [prefix/etc].
--sharedstatedir=dir	Datos independientes de arquitectura modificables en dir [prefix/com].
--localstatedir=dir	Datos de una máquina única modificable en dir[prefix/var].
--libdir=dir	Bibliotecas de código de objetos en dir[eprefix/lib].
--includedir=dir	Archivos de cabecera C en dir[prefix/include].
--oldincludedir=dir	Archivos de cabecera para GCC en dir [/usr/include].
--infodir=dir	Documentación de información en dir[prefix/info].
--mandir=dir	Documentación man en dir[prefix/man].
--srcdir=dir	Encuentra fuentes en dir[configure dir or ...].
--program-prefix=prefix	Utiliza <i>prefix</i> para instalar nombres de programas.
--program-suffix=suffix	Añade <i>suffix</i> para instalar los nombres de programas.
--program-transform-name=program	Ejecuta el programa editor de flujo en la instalación de nombres de programas.
--build=build	Configuración para construir con <i>build</i> .
--host=host	Configuración para <i>host</i> .
--target=target	Configuración para TARGET [TARGET=HOST].

Opción	Significado
--disable-feature	No incluir FEATURE (igual que --enable-FEATURE=no).
--enable- feature[=arg]	Incluir <i>feature</i> [arg=yes].
--with-package[=arg]	Utilizar <i>package</i> [arg=yes].
--without-package	No utilizar (igual que --with- package=no).
--x-includes=dir	Hay X archivos include en <i>dir</i> .
--x-libraries=dir	Hay X archivos library en <i>dir</i> .
--with-optim=flag	Obsoleto (utiliza la variable de entorno OPTIM ).
--with-port=port	Puerto al que escuchar (el puerto por defecto es el 80).
--enable-debug	Enciende las advertencias de debugging y de compilación en tiempo de ejecución.
--enable-maintainer-mode	Enciende las advertencias de debugging y de compilación en tiempo de ejecución.
--enable-layout=layout	Permite una distribución de directorio.
--enable-modules=module-list	Permite uno o más módulos.
--enable-mods-shared=module-list	Permite uno o más módulos como módulos compartidos.
--disable-access	Inutiliza el control de acceso basado en host.
--disable-auth	Inutiliza el control de acceso basado en usuario.
--enable-auth-anon	Permite acceso a usuarios anónimos.
--enable-auth-dbm	Permite acceso a bases de datos basado en DMB.
--enable-auth-db	Permite acceso a bases de datos basado en DB.
--enable-auth-digest	Permite la autentificación por digestión RFC2617.
--enable-file-cache	Permite caché de archivos.
--enable-dav-fs	Permite manejo de protocolo DAV.
--enable-dav	Permite manejo de protocolo WebDAV.
--enable-echo	Permite servidor ECHO.

Opción	Significado
--enable-charset-lite	Permite la traducción de conjuntos de caracteres.
--enable-cache	Permite caching dinámico de archivos.
--enable-disk-cache	Permite el módulo de caching de disco.
--enable-ext-filter	Permite módulos de filtro externo.
--enable-case-filter	Permite filtro de conversión de mayúsculas.
--enable-generic-hook	Ejemplo permitido de exportación de entradas.
-export	
--enable-generic-hook	Ejemplo permitido de importación de entradas.
-import	
--enable-optional-fn	Ejemplo permitido de importación opcional de una función.
-import	
--enable-optional-fn	Ejemplo permitido de exportación opcional de una función.
-export	
--disable-include	Inutiliza Server-Side Includes.
--disable-http	Inutiliza el manejo de protocolo HTTP.
--disable-mime	Inutiliza la integración de la extensión de archivos MIME.
--disable-log-config	Inutiliza la configuración de registro.
--enable-vhost-alias	Permite el módulo de alojamiento.
--disable-negotiation	Inutiliza la negociación de contenido.
--disable-dir	Inutiliza el manejo de solicitudes.
--disable-imap	Inutiliza la integración interna de imágenes.
--disable-actions	Inutiliza el lanzamiento de solicitudes.
--enable-speling	Permite la corrección ortográfica de las URL habituales.
--disable-userdir	Inutiliza la integración de las solicitudes de usuarios.
--disable-alias	Inutiliza la traducción de solicitudes.
--enable-rewrite	Permite la reescritura de las URL.
--disable-so	Inutiliza la capacidad DSO.
--enable-so	Permite la capacidad DSO.
--disable-env	Borra/asigna variables ENV.
--enable-mime-magic	Determina automáticamente el tipo MIME.

Opción	Significado
--enable-cern-meta	Permite meta archivos de tipo CERN.
--enable-expires	Permite el control de cabeceras Expires.
--enable-headers	Permite el control de cabeceras HTTP.
--enable-usertrack	Permite seguimiento de la sesión de usuario.
--enable-unique-id	Permite solicitudes de un solo ID.
--disable-setenvif	Inutiliza las variables ENV en las cabeceras.
--enable-tls	Permite soporte TLS/SSL.
--with-ssl	Utiliza una instalación especial de la biblioteca SSL.
--with-mpm=MPM	Elige el modelo de proceso para Apache para utilizar: MPM={beos threaded prefork spmt_os2 perchild}.
--disable-status	Monitoriza procesos / hilos.
--disable-autoindex	Inutiliza la lista de directorios.
--disable-asis	Inutiliza los tipos de archivo As-is.
--enable-info	Permite información del servidor.
--enable-suexec	Asigna UID y GID a los procesos engendrados.
--disable-cgid	Inutiliza el soporte del demonio CGI.
--enable-cgid	Permite el soporte del demonio CGI.
--disable-cgi	Inutiliza el soporte de scrips CGI.
--enable-cgid	Permite el soporte de scripts CGI.
--enable-shared[=pkgs]	Construye bibliotecas compartidas [default=no].
--enable-static[=pkgs]	Construye bibliotecas estáticas [default=yes].
--enable-fast-install[=pkgs]	Optimiza la instalación rápida [default=yes].
--with-gnu-ld	Adopta la utilización de compilador C GNU ID [default=no].
--disable-libtool-lock	Evita el bloqueo (posiblemente rompe la construcción paralela).
--with-program-name	Alterna nombres ejecutables.
--with-suexec-caller	Usuario que puede realizar llamadas SuExec.

Opción	Significado
--with-suexec-userdir	Subdirectorio del usuario.
--with-suexec-docroot	Directorio raíz SuExec.
--with-suexec-uidmin	Mínimo UID permitido.
--with-suexec-gidmin	Mínimo GID permitido.
--with-suexec-logfile	Asigna el fichero de registro.
--with-suexec-safepath	Asigna el safepath.
--with-suexec-umask	Una máscara para el proceso suexec.

La mayoría de estas opciones no son necesarias para la gran parte de los sitios. Normalmente, lo único que necesita es especificar la opción `--prefix` y cualquier otra opción necesaria para permitir o no uno u otro módulo. Por ejemplo, imagine que no quiere instalar el módulo CGI en su sistema. Puede ejecutar el script `configure` utilizando las opciones `--disable-cgi` `--disable-cgid` para inutilizar el soporte CGI. Del mismo modo, para inutilizar el soporte Server-Side Include (SSI) puede utilizar la opción `--disable-include`.

Una vez que ha configurado el Apache con el script `configure`, puede utilizar el script `config.status` en lugar del script `configure` para las configuraciones siguientes. Utilizando el script `config.status`, puede reutilizar su configuración previa y añadir o quitar opciones. Por ejemplo, imagine que configuró Apache con la siguiente línea de comando:

```
./configure --prefix=/usr/local/apache --disable-cgi --disable-cgid
```

y unos días más tarde decide inutilizar las SSI. Puede utilizar ahora:

```
./config.status --disable-include
```

Cuando vuelva a compilar Apache, los módulos CGI no estarán incluidos, porque `./config.status` almacena las opciones que especificó al utilizar el script `configure` antes.

Si quiere empezar bien, utilice siempre `configure`.

## Opciones avanzadas de configuración para sitios con mucho tráfico

Si ejecuta Apache en un servidor con mucho tráfico en el que se solicitan cientos de solicitudes por segundo, debería cambiar el límite de hardware por defecto asignado en el módulo MPM que ha elegido. Ver el capítulo 1 para obtener los detalles sobre los módulos MPM. Los límites de hardware por defecto que

puede cambiar son HARD\_SERVER\_LIMIT y HARD\_THREAD\_LIMIT. El HARD\_SERVER\_LIMIT asigna el número máximo de hijos que el servidor Apache puede producir. HARD\_THREAD\_LIMIT asigna el número total de hilos que Apache puede crear dentro de sus hijos. La tabla 2.3 muestra los límites por defecto y dónde puede encontrarlos.

**NOTA:** La etiqueta %APACHE\_SOURCE% mencionada en la tabla se refiere al directorio de distribución de fuente Apache. Por ejemplo, si ha extraído distribución de fuente Apache en el directorio /usr/local/src/httpd\_version, entonces reemplace %APACHE\_SOURCE% con /usr/local/src/httpd\_version para localizar el archivo cabecera (include) C adecuado.

**Tabla 2.3.** Limitaciones de hardware para los distintos módulos MPM de Apache

MPM	Opciones límite	Valor por defecto	Anotaciones
Threaded	HARD_SERVER_LIMIT para el modo threaded	8	Archivo de cabecera C (include): %APACHE_SOURCE%/server/mpm/threaded/mpm_default.h #ifndef NO_THREADS #define HARD_SERVER_LIMIT 256 #endif #ifndef HARD_SERVER_LIMIT #define HARD_SERVER_LIMIT 8 #endif
Threaded	HARD_THREAD_LIMIT para el modo threaded	64	Archivo de cabecera C (include): %APACHE_SOURCE%/server/mpm/threaded/mpm_default.h
Prefork	HARD_SERVER_LIMIT	256	Archivo de cabecera C (include): %APACHE_SOURCE%/server/mpm/prefork/mpm_default.h #ifndef HARD_SERVER_LIMIT #define HARD_SERVER_LIMIT 256 #endif
Perchild	HARD_SERVER_LIMIT	8	Archivo de cabecera C (incluido): %APACHE_SOURCE%/server/mpm/perchild/mpm_default.h #ifndef HARD_SERVER_LIMIT #define HARD_SERVER_LIMIT 8 #endif
Perchild	HARD_THREAD_LIMIT	64	Archivo de cabecera C (include): %APACHE_SOURCE%/server/mpm/perchild/mpm_default.h #ifndef HARD_SERVER_LIMIT #define HARD_SERVER_LIMIT 8 #endif

MPM	Opciones límite	Valor por defecto	Anotaciones
Winnt	HARD_SERVER_LIMIT 1		Archivo de cabecera C (include): %APACHE_SOURCE%/server/mpm/winnt/ mpm_default.h #define HARD_SERVER_LIMIT 1 Esta asignación no se puede cambiar. Podemos, sin embargo, cambiar el conteo de hilos de hardware
Winnt	HARD_THREAD_LIMIT 4096		Archivo de cabecera C (include): APACHE_SOURCE/server/mpm/winnt/ mpm_default.h #ifndef HARD_THREAD_LIMIT #define HARD_THREAD_LIMIT 4096 #endif

**ADVERTENCIA:** Cuando cambie HARD\_SERVER\_LIMIT o HARD\_THREAD\_LIMIT a uno superior que el que tenemos por defecto, asegúrese de que tiene los recursos del sistema apropiados. Por ejemplo, el cambiar HARD\_SERVER\_LIMIT a 1024 en el MPM prefork le permitirá crear 1024 procesos hijo Apache instruyendo a Apache para que cree todos esos hijos utilizando las directivas StartServers, MinSpareServers y MaxSpareServers.

Sin embargo, si su sistema no tiene suficiente memoria, entonces cambiar el límite de hardware a un valor más alto no beneficiará demasiado. Recuerde que cuanto más altos sean los límites, más recursos necesitará. Además, tendrá que aumentar el número de descriptores de archivos que su sistema permite para un solo usuario. En los sistemas Linux y Unix debería determinar cuál es (y posiblemente asignar) el límite de descriptores de archivos utilizando el comando ulimit.

Además, no olvide que la directiva MaxClients asigna el límite en el número de procesos hijo que se crearán para servir solicitudes. Cuando el servidor se construye sin threaded, no se puede servir a un número de clientes mayor que éste de forma simultánea. Para configurar más de 256 clientes, debe editar la entrada HARD\_SERVER\_LIMIT en mpm\_default.h y volver a compilar.

## Compilar e instalar Apache

Una vez que tiene configurada la fuente de Apache utilizando el script configure ha de seguir los siguientes pasos para compilar e instalar Apache:

1. Ejecute el comando make para compilar la fuente.
2. Ejecute el comando make install para instalar httpd y soportar archivos en el directorio que determine utilizando la opción --prefix.

3. Diríjase al directorio de instalación y navegue por el directorio. Verá subdirectorios del tipo bin cgi-bin, conf, htdocs, icons, include, lib y logs. Por ejemplo, si utiliza prefix=/usr/local/apache con el script configure durante la configuración del árbol fuente, make install creará la siguiente estructura de directorio:

```
/usr/local/apache
|
+---include
+---lib
+---bin
+---conf
+---htdocs
|   |
|   +--manual
|       |
|       +--developer
|       +--howto
|       +--images
|       +--misc
|       +--mod
|       +--platform
|       +--programs
|       +--search
|       +--vhosts
|
+---icons
|   |
|   +--small
|
+---logs
+---cgi-bin
```

La siguiente lista le ofrece una breve descripción de cada uno de los directorios de la estructura de directorios:

- **Include:** contiene todos los archivos cabecera (include) que sólo son necesarios si desarrolla aplicaciones Web que integran con Apache o si quiere utilizar software de terceras partes con Apache. En un servidor de producción puede eliminar este directorio.
- **Lib:** aloja los archivos de la biblioteca Apache Portable Run-Time (APR), los archivos que son necesarios para ejecutar Apache y otras utilidades de soporte como ab.
- **Bin:** contiene los programas que se muestran en la tabla 2.4.
- **Conf:** aloja los archivos de configuración de Apache. Contiene los archivos de la tabla 2.5.

- **Htdocs:** este es el directorio raíz de documentos para el servidor principal de Apache. El archivo `httpd.conf` asigna la directiva `DocumentRoot` a este directorio. Aprenderá cómo asignar su propio directorio raíz de documentos en el capítulo 3. Por defecto, el directorio `htdocs` también tiene el manual de instalación de Apache en un subdirectorio.
- **Icons:** se utiliza para almacenar varios iconos de Apache necesarios para desplegar de forma dinámica la lista construida de directorios.
- **Logs:** se utiliza para almacenar los registros del servidor de Apache, el demonio CGI basado en socket (`cgisock`) y el archivo PID (`httpd.pid`). Aprenderá a cambiar la ruta de registro en el capítulo 3.
- **Cgi-bin:** el directorio de script CGI por defecto, que se asigna utilizando la directiva `ScriptAlias` en `httpd.conf`. Por defecto, Apache tiene dos scripts CGI sencillos `printenv` y `test-cgi`. Cada uno de estos scripts imprime variables de entorno CGI cuando realiza solicitudes mediante `http://server_name/cgi-bin/script_name URL`. Estos scripts son necesarios para determinar si está funcionando la configuración CGI.

**ADVERTENCIA:** Se recomienda eliminar los scripts `printenv` y `test-cgi` una vez que se encuentre funcionando la configuración CGI. No es una buena idea tener un script que muestra información sobre su sistema a cualquiera. Cuanto menos sepa el resto del mundo sobre el modo de funcionar de su sistema, más seguro será éste.

La tabla 2.4 proporciona una lista de programas que puede encontrar en el directorio `bin`.

**Tabla 2.4.** Programas Apache en el directorio `bin`

Programas Apache	Definición
Ab	Este es el programa <code>apachebench</code> . Le sirve como punto de referencia al servidor Apache. Ver el capítulo 22 para obtener más información sobre este programa.
Apachectl	Es un script de gran utilidad que le permite iniciar, reiniciar y parar el servidor Apache. Ver el capítulo 3 para obtener más información sobre este script.
apxs	Esta es una herramienta para construir e instalar módulos de extensión de Apache. Permite construir módulos DSO que se pueden utilizar en Apa-

Programas Apache	Definición
	che utilizando el módulo mod_so. Para más información sobre este programa, ver <a href="http://your_server_name/manual/programs/apxs.htm">http://your_server_name/manual/programs/apxs.htm</a> .
htdigest	Este programa crea y actualiza la información de autenticación de usuarios cuando se utiliza la autenticación por digestión de mensajes (MD5). Para obtener más información sobre este programa ver <a href="http://your_server_name/manual/programs/htdigest.html">http://your_server_name/manual/programs/htdigest.html</a> .
htpasswd	Este programa se utiliza para crear y actualizar la información de autenticación de usuarios en autenticación HTTP básica. Ver el capítulo 7 para obtener más detalles.
httpd	Este es el programa del servidor Web de Apache.
logresolve	Este programa convierte (resuelve) las direcciones IP de un archivo de registro a nombres de host. Ver el capítulo 8 para obtener los detalles.
Rotatelogs	Este programa alterna los archivos de registro de Apache cuando alcanzan un tamaño determinado. Ver el capítulo 8 para obtener los detalles.

La tabla 2.5 muestra la lista de contenidos en el directorio config.

**Tabla 2.5.** Contenido del directorio config de Apache

Archivo de configuración	Definición
httpd.conf	Este es el archivo de configuración de Apache.
httpd-std.conf	Esta es la copia de prueba del archivo httpd.conf, que Apache no necesita. Para los usuarios nuevos de Apache, este archivo puede actuar recuperando el conf por defecto.
highperformance.conf	Este es un archivo de prueba de configuración que muestra algunos consejos para configurar Apache para alto rendimiento.
highperformance-std.conf	Esta es una copia de prueba del archivo highperformance.conf, que Apache no necesita.
Magic	Este archivo almacena los datos mágicos del módulo mod_mime_magic de Apache.

Archivo de configuración	Definición
mime.types	Este archivo se utiliza para decidir qué tipo de cabecera MIME se envía al cliente para un archivo determinado. Para obtener más información sobre los tipos MIME, por favor lea RFC 2045, 2046, 2047, 2048 y 2077. El registro los Internet media-types se encuentra en el sitio <a href="ftp://ftp.iana.org/in-notes/iana/assignments/media-types">ftp://ftp.iana.org/in-notes/iana/assignments/media-types</a> .

## Instalar Apache desde los paquetes binarios RPM

Puede bajar los binarios de Apache apropiados para su sistema de [www.apache.org/dist/httpd/binaries](http://www.apache.org/dist/httpd/binaries). Baje la última versión y extraiga el archivo comprimido en un directorio temporal. Para saber cómo instalar los binarios en su plataforma, debe leer el archivo `install.bindist`, incluido en cada distribución del binario.

Si desea instalar el paquete RPM (Red Hat Package Management) de Apache en su sistema Linux, haga lo siguiente:

1. Diríjase al sitio <http://rpmfind.net> y busque la cadena de caracteres Apache para localizar los paquetes RPM. Localice la última versión de la distribución del RPM y bájesele.
2. Ejecute el comando `rpm -ivh apache_rpm_package.rpm` para instalar el paquete. Por ejemplo, para instalar el `apache-2.0.4-i386.rpm` para el sistema Red Hat Linux (Intel), ejecute el comando `rpm -ivh apache-2.0.4-i386.rpm`.

## Mantenerse al día en el desarrollo de Apache

Se puede preguntar si mientras está instalando la fuente o los binarios de Apache que ha bajado está saliendo una nueva versión de Apache o quizás hay disponible un parche de seguridad. El software cambia rápidamente en los tiempos que corren y siempre aparece una actualización detrás de otra. Lo cual es bueno, pero no siempre es fácil mantenerse al día si tenemos un trabajo que hacer. Existen dos recursos de Apache que debería tener en cuenta:

- **ApacheToday:** este es el mejor sitio Web en el mundo de noticias de Apache. Puede obtener todas las noticias de Apache que quiera en

[www.apachetoday.com](http://www.apachetoday.com). Utilizando su artículo Your Apache Today, puede filtrar contenido de noticias y obtener exactamente lo que más le interesa. Puede asistir también a los eventos que desee de los que se anuncian en estas noticias.

- **ApacheWeek:** se puede suscribir (gratis) al recurso de Apache llamado Apache Week, y le mandarán por correo electrónico todas las noticias de Apache. Puede encontrar el sitio Web The Apache Week en [www.apacheweek.com](http://www.apacheweek.com). Se trata de un gran recurso informativo para los administradores de Apache que quieran estar realmente informados. Puede leer además muchos artículos de gran utilidad sobre cómo sacar el máximo partido de su servidor. Le recomiendo visitar este sitio Web.





# 3 Preparar y ejecutar Apache

---

## En este capítulo

1. Comprobamos los aspectos básicos del servidor Apache.
2. Iniciamos, paramos y reiniciamos el servidor Apache.
3. Probamos un servidor Apache en funcionamiento.

En el último capítulo, aprendió a compilar y a instalar el servidor Web Apache en su sistema Unix. Ahora está listo para prepararlo y ejecutarlo. Este capítulo cubre los detalles básicos de configuración.

## Configurar Apache

Por defecto, Apache lee un solo archivo de configuración llamado `httpd.conf`. Cada distribución de código fuente de Apache viene con un conjunto de archivos de configuración de ejemplo. En la distribución estándar de código fuente de Apache, puede encontrar un directorio llamado `conf`, que contiene archivos de configuración de ejemplo con la extensión `dist`.

Antes de modificar este archivo es necesario que cree una copia backup del original.

El archivo `httpd.conf` contiene dos tipos de información: comentarios y directivas de servidores. Las líneas que comienzan con un carácter `#` son tratadas como líneas de comentario; estos comentarios no tienen significado para el software del servidor, sino que son documentación para el administrador del servidor. Puede añadir tantos comentarios como desee; el servidor simplemente ignora todos los comentarios cuando analiza el archivo. Exceptuando los comentarios y las líneas en blanco, el servidor trata el resto de las líneas bien como directivas completas o bien como directivas parciales. Una es como un comando para el servidor. Le dice al servidor que realice una tarea determinada de un modo en concreto. Mientras editamos el archivo `httpd.conf`, ha de tomar ciertas decisiones relacionadas con cómo quiere que se comporte el servidor. En las siguientes secciones, aprenderá qué significan estas decisiones y cómo puede utilizarlas para personalizar su servidor.

Puede encontrar una explicación en profundidad sobre todas las directivas principales más adelante en este libro.

El listado 3.1 muestra el `httpd.conf` creado por defecto en el directorio `conf` de su instalación de Apache. La mayor parte de los comentarios se han eliminado y el código se ha editado resumido por cuestiones de brevedad.

#### Listado 3.1. `httpd.conf` creado por defecto desde `httpd.conf-dist`

```
### Sección 1: Entorno global
ServerRoot "/usr/local/apache"

PidFile logs/httpd.pid

<IfModule !perchild.c>
    ScoreBoardFile logs/apache_runtime_status
</IfModule>

Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

<IfModule prefork.c>
    StartServers      5
    MinSpareServers  5
    MaxSpareServers  10
    MaxClients       20
    MaxRequestsPerChild  0
</IfModule>

<IfModule threaded.c>
    StartServers      3
    MaxClients        8
```

```

MinSpareThreads      5
MaxSpareThreads     10
ThreadsPerChild     25
MaxRequestsPerChild  0
</IfModule>

<IfModule perchild.c>
    NumServers          5
    StartThreads        5
    MinSpareThreads     5
    MaxSpareThreads     10
    MaxThreadsPerChild  20
    MaxRequestsPerChild 0
</IfModule>

### Sección 2: Configuración principal del servidor
Port 80
User nobody
Group #-1
ServerAdmin you@your.address

# Añadido
ServerName www.domain.com

DocumentRoot "/usr/local/apache/htdocs"

<Directory />
    Options FollowSymLinks
    AllowOverride None

</Directory>

<Directory "/usr/local/apache/htdocs">
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

UserDir public_html
DirectoryIndex index.html
AccessFileName .htaccess

<Files ~ "\.ht">
    Order allow,deny
    Deny from all
</Files>

UseCanonicalName On
TypesConfig conf/mime.types
DefaultType text/plain

<IfModule mod_mime_magic.c>

```

```

MIMEMagicFile conf/magic
</IfModule>

HostnameLookups Off
ErrorLog logs/error_log
LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
CustomLog logs/access_log common
ServerSignature On
Alias /icons/ "/usr/local/apache/icons/"

<Directory "/usr/local/apache/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

ScriptAlias /cgi-bin/ "/usr/local/apache/cgi-bin/"

<Directory "/usr/local/apache/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>

IndexOptions FancyIndexing VersionSort
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

AddIconByType (TXT,/icons/text.gif) text/*
# Hay muchas más directivas AddIconByType en el
# httpd.conf por defecto pero se han borrado
# para simplificar.

AddIcon /icons/binary.gif .bin .exe
# Hay muchas más directivas AddIcon en
# httpd.conf por defecto pero se han borrado para
# simplificar.

DefaultIcon /icons/unknown.gif
ReadmeName README
HeaderName HEADER
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
AddEncoding x-compress Z
AddEncoding x-gzip gz tgz

AddLanguage da .dk
AddLanguage nl .nl

```

```

AddLanguage en .en
# Hay muchas más directivas AddLanguage s en el
# httpd.conf por defecto pero se han borrado para
# simplificar.

LanguagePriority en da nl et fr de el it ja kr no pl pt pt-br
ltz ca es sv tw

AddDefaultCharset ISO-8859-1
AddCharset ISO-8859-1 .iso8859-1 .latin1
AddCharset ISO-8859-2 .iso8859-2 .latin2 .cen
# Hay muchas más directivas AddCharset en
# httpd.conf por defecto pero se han borrado para
# simplificar.

AddType application/x-tar .tgz
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-
response-1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0

### Sección 3: Hosts Virtuales
# No hay servidores virtuales definidos en este capítulo.

```

Como el objetivo de este capítulo es preparar y ejecutar el servidor con la configuración mínima, el capítulo no ofrece detalles en profundidad sobre las opciones de configuración. Puede aprender todos los detalles en este libro.

El archivo `httpd.conf` en realidad no tiene ninguna sección delimitadora. La figura 3.1 le ayudará a entender mejor el archivo de configuración.

El lado izquierdo de la figura muestra cómo podemos visualizar el `httpd.conf` por defecto en nuestra mente. Hay directivas de configuración que crean el entorno global del servidor que se aplica a todo; hay opciones de configuración que se aplican al sitio Web principal (por defecto) de los servidores Apache, y hay directivas de configuración que sólo se aplican a host virtuales opcionales.

Como Apache utiliza un solo archivo de configuración, un sitio con muchos host virtuales tendrá un gran archivo y la gestión de la configuración se hará muy engorrosa. Este es el motivo por el cual muestro un modo de desglosar el archivo `httpd.conf`. Pero discutiré esta aproximación (lo que se encuentra evidentemente en el lado derecho de la figura anterior) en el capítulo 6. Tal y como dije antes, nos vamos a centrar en la preparación y en la ejecución.

**NOTA:** Siempre que me refiera a `%directive%`, me estoy refiriendo al valor del conjunto de directivas en el archivo de configuración. Por ejemplo, si una directiva llamada `ServerAdmin` está fijada con el valor

kabir@domain.com, entonces una referencia a %ServerAdmin% significará "kabir@domain.com". Por lo tanto, si le pido que cambie %ServerAdmin%, le estoy pidiendo que cambie la dirección de la cuenta de correo.

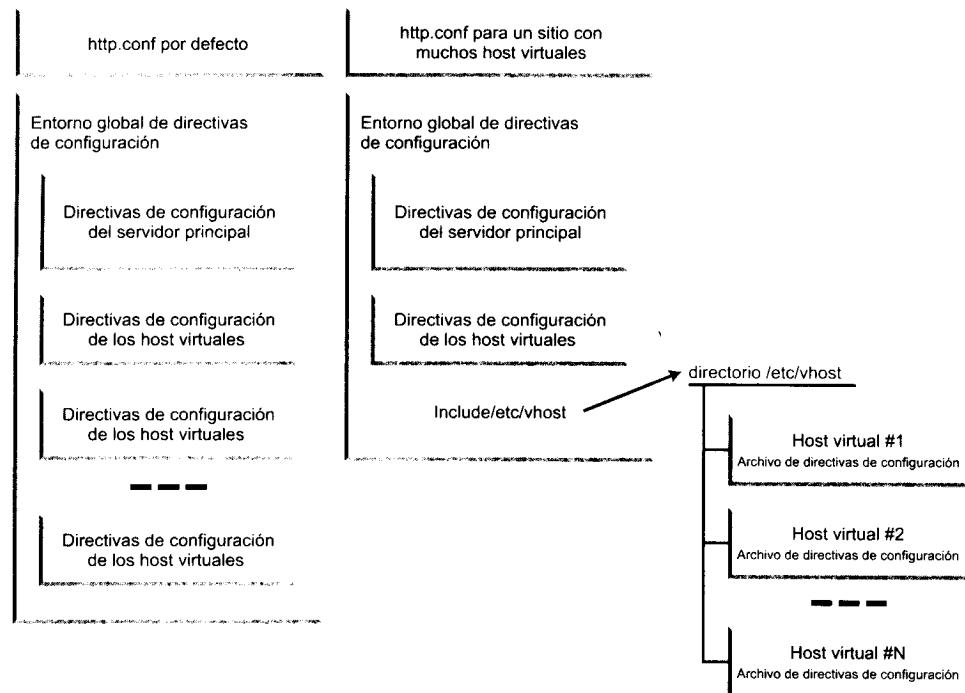


Figura 3.1. Los segmentos de configuración de httpd.conf

## Configurar el entorno global para Apache

Las directivas discutidas en esta sección crean el entorno global para el servidor Apache. Las directivas se discuten en el mismo orden en el que aparecen en el archivo httpd.conf. La primera directiva es ServerRoot, y es la siguiente:

`ServerRoot "/usr/local/apache"`

Esta directiva especifica el directorio de máximo nivel del servidor Web. El directorio especificado no es en el que se guarda el contenido Web. En realidad es un directorio que normalmente tiene los siguientes subdirectorios:

```
{ServerRoot Directory}
|
|---bin
|---conf
|---htdocs
```

```
| ----htdocs/
|   |
|   +--manual
|       |----developer
|       |----howto
|       |----images
|       |----misc
|       |----mod
|       |----platform
|       |----programs
|       |----search
|       +----vhosts
|
| ----icons
|   |
|   +--small
|
| ----logs
| ----cgi-bin
+----include
```

/usr/local/apache es el directorio padre para todos los archivos relacionados con el servidor. El valor por defecto para ServerRoot es fijo para cualquiera de las opciones elegidas en --prefix durante la configuración del código fuente utilizando el script configure. Por defecto, al hacer que se ejecute el comando install durante la instalación del servidor se copian todos los binarios del servidor en %ServerRoot%/bin, los archivos de configuración en %ServerRoot%/conf, y así sucesivamente.

**NOTA:** Debe cambiar únicamente el valor de esta directiva si ha movido manualmente el directorio completo desde el lugar de instalación a otra localización. Por ejemplo, si simplemente ejecuta cp -r /usr/local/apache /home/apache y quiere configurar el servidor Apache para que trabaje desde la nueva localización, ha de cambiar esta directiva a ServerRoot /home/apache. Tenga en cuenta que en ese caso debe cambiar también otras referencias a la directiva de /usr/local/apache a /home/apache.

Además tenga en cuenta que cada vez que vea un nombre de directorio relacionado, Apache le añadirá el prefijo %ServerRoot% a la ruta para construir la verdadera ruta. Verá un ejemplo en la siguiente sección.

## PidFile

La directiva PidFile fija la ruta del archivo PID (proceso ID). Por defecto, esta fijada en logs/httpd.pid, que se traduce en %ServerRoot%/logs/httpd.pid (es decir, /usr/local/apache/logs/httpd.pid). Cuando

quiera encontrar el PID del proceso principal de Apache que ejecuta como root y produce procesos hijo, puede ejecutar el comando `cat %serverroot/logs/httpd.pid`. No olvide reemplazar `%ServerRoot%` con el valor apropiado.

**ADVERTENCIA:** Si cambia el valor `%PidFile%` para dirigirse a una localización diferente, asegúrese de que el directorio en el que se encuentra el archivo `httpd.pid` no es sobreescribible por nadie excepto por el usuario `root`, por razones de seguridad.

## ScoreBoardFile

ScoreBoardFile se encuentra encapsulado dentro de una condición if utilizando el contenedor `<IfModule . . .>`, tal y como se muestra a continuación:

```
<IfModule !perchild.c>
    ScoreBoardFile logs/apache_runtime_status
</IfModule>
```

Le dice a Apache que asigne ScoreBoardFile al archivo `%ServerRoot%/logs/apache_runtime_status` sólo si ha elegido un módulo multiproceso (MPM) distinto al `perchild`. Como el MPM por defecto para la mayoría de los sistemas operativos, incluido Linux, es threaded en lugar de `perchild`, la condición if será verdadera y Apache asignará la directiva `scoreboardfile`. Esta directiva se utiliza para dirigirse a un archivo, que se emplea para intercambiar información del estado en tiempo de ejecución entre los procesos Apache. Si tiene un disco RAM, debería considerar colocar este archivo en el disco RAM para aumentar el rendimiento. En la mayoría de los casos, deberá dejar únicamente esta directiva.

## Timeout, KeepAlive, MaxKeepAliveRequests y KeepAliveTimeout

Timeout fija el tiempo en segundos. Debe conservarse el valor por defecto. Las tres directivas siguientes KeepAlive, MaxKeepAliveRequests y KeepAliveTimeout se utilizan para controlar el comportamiento keep-alive (mensajes intercambiados por los routers que informan que una sesión sigue activa) del servidor. No tiene que cambiarlo.

## Contenedores IfModule

Apache utilizará uno de los siguientes contenedores `<IfModule . . .>` basados en el MPM que elija. Por ejemplo, si configura Apache utilizando el modo por defecto de MPM (threaded) en un sistema Linux, entonces utilizaremos el contenedor `<IfModule . . .>` siguiente:

```
<IfModule threaded.c>
    StartServers          3
    MaxClients           8
    MinSpareThreads      5
    MaxSpareThreads     10
    ThreadsPerChild     25
    MaxRequestsPerChild  0
</IfModule>
```

Sin embargo, si elige `--with-mpm=prefork` durante la configuración de la fuente utilizando el script de configuración, entonces utilizaremos el contenedor `<IfModule . . .>` siguiente:

```
<IfModule prefork.c>
    StartServers          5
    MinSpareServers       5
    MaxSpareServers      10
    MaxClients           20
    MaxRequestsPerChild  0
</IfModule>
```

Del mismo modo, la opción `--with-mpm=perchild` fuerza a Apache a utilizar el contenedor `<IfModule . . .>`

## **Directivas para el comportamiento MPM threaded (comportamiento MPM por defecto)**

Si ha seguido mi consejo del capítulo anterior, no habrá cambiado el comportamiento MPM por defecto durante la compilación de la fuente, por lo que las directivas que ha de considerar se discuten a continuación.

**NOTA:** Si ha cambiado el MPM por defecto, puede encontrar información detallada sobre las directivas necesarias para el MPM que haya elegido en el capítulo 4.

### **StartServers**

`StartServers` le dice a Apache que inicie tres servidores Apache al arrancar. Puede iniciar más servidores si quiere, pero Apache es muy bueno aumentando el número de procesos hijo según sus necesidades basadas en la carga. Por tanto, no es necesario cambiar esta directiva.

### **MaxClients**

En el modo MPM threaded por defecto, el número total de solicitudes que Apache puede procesar es `%MaxClients% x %ThreadsPerChild%`. Por lo que, dado que el número por defecto para `MaxClients` es 8 y el número por

defecto para `ThreadsPerChild` es 25, el número por defecto máximo para respuestas simultáneas es 200 (es decir, 8 veces 5). Si utiliza el MPM preforking, el máximo de solicitudes está limitado a `%MaxClients%`. El número máximo por defecto de 200 solicitudes simultáneas debe funcionar adecuadamente en la mayor parte de los sitios, por lo que debemos dejar los valores fijados por defecto.

## **MinSpareThreads**

La directiva `MinSpareThreads` determina el número mínimo de hilos parados. Estos hilos sobrantes se utilizan para servir solicitudes y los hilos de sobra se crean para mantener el mínimo tamaño del grupo de hilos parados. Puede dejar los valores asignados por defecto.

## **MaxSpareThreads**

La directiva `MaxSpareThreads` determina el número máximo de hilos parados; deje el valor por defecto tal y como está. En el modo threaded, que es el modo por defecto, Apache mata los procesos hijo para controlar la cuenta de hilos mínimos y máximos.

## **ThreadsPerChild**

Esta directiva determina la cantidad de hilos que se crean por cada proceso hijo.

**NOTA:** Si está ejecutando Apache en un sistema Windows, asigne `ThreadsPerChild` al máximo número de respuestas simultáneas que quiera manejar, porque en esta plataforma sólo hay un proceso hijo y es dueño de todos los hilos.

## **MaxRequestPerChild**

La última directiva para el entorno global es `MaxRequestPerChild`, la cual fija el número de solicitudes que un proceso hijo puede servir antes de ser asesinado. El valor por defecto de cero hace que el proceso hijo sirva solicitudes siempre. No me gusta este valor por defecto, porque permite que los procesos Apache consuman poco a poco gran cantidad de memoria cuando un script `mod_perl` defectuoso, o incluso un módulo defectuoso de Apache, filtra memoria. Por eso, prefiero asignar el valor 30.

**TRUCO:** Si no tiene pensado ejecutar ningún otro módulo Apache o scripts `mod_perl`, puede dejar los valores por defecto o asignar cualquier número razonable. Una asignación de 30 asegura que ese proceso hijo es asesinado después de procesar 30 solicitudes. Por supuesto, se crea un nuevo proceso hijo cuando es necesario.

# Configurar el servidor principal

El servidor principal se refiere al sitio Web por defecto de los servidores Apache. Este es el sitio que aparece cuando ejecuta Apache y utiliza la dirección IP del servidor o el nombre del host en un navegador Web.

## Puerto

La primera directiva en esta sección es la directiva `Port`, que asigna el puerto TCP que Apache escucha para conectarse. El puerto HTTP estándar tiene un valor de 80 por defecto. Si cambia este valor por otro número, como el 8080, únicamente puede acceder al servidor utilizando una URL del tipo `http://hostname:8080/`. Debe especificar el número de puerto en la URL si el servidor no se ejecuta en un puerto estándar.

Existen muchas razones para ejecutar Apache en puertos que no son estándar, pero la mejor que puedo dar es que no tiene permiso para ejecutar Apache en el puerto estándar HTTP. Como usuario no-raíz puede ejecutar Apache únicamente en puertos superiores a 1024.

Una vez que ha decidido ejecutar Apache utilizando un puerto, necesita decirle a Apache qué nombres de usuario y de grupos hay.

## Directivas de usuarios y grupos

Las directivas `User` y `Group` le dicen a Apache los nombres de usuario (UID) y de grupo (GID) que tiene que utilizar. Estas dos directivas son muy importantes por razones de seguridad. Cuando el proceso principal del servidor Web lanza un proceso hijo para satisfacer una solicitud, cambia el UID y el GID del hijo de acuerdo con los valores fijados para estas directivas. Remítase a la figura 3.1 para ver cómo el proceso principal del servidor Web, que escucha la conexión, se ejecuta como un proceso raíz de usuario, y cómo el proceso hijo se ejecuta como procesos usuario / grupo distintos. Si los procesos hijo se están ejecutando como procesos raíz de usuario, se abrirá un agujero para ataques potenciales de hackers. Permitir la capacidad de interaccionar con un proceso raíz de usuario maximiza la posibilidad de una brecha de seguridad en el sistema; por tanto, no está recomendado. Lo que realmente recomiendo es que elija ejecutar los procesos hijo del servidor como un usuario de escaso privilegio que pertenece a un grupo de bajo privilegio. En la mayor parte de los sistemas Unix, el usuario denominado `nobody` (normalmente GID -1) y el grupo denominado `nogroup` (normalmente GID -1) tienen bajo privilegio. Debe consultar los archivos `/etc/group` y `/etc/passwd` para determinar estos ajustes.

Si pretende ejecutar el servidor Web principal como un usuario no-raíz (normal), no podrá cambiar el UID y el GID de los procesos hijo, porque únicamente los procesos de un usuario `root` pueden cambiar el UID o el GID de otros procesos. Por lo tanto, si quiere ejecutar su servidor principal como el usuario llamado `ironsheik`, entonces todos los procesos hijo tienen los mismos privi-

legios que iron sheik. Del mismo modo, cualquier ID de grupo que tenga será también el ID de grupo para el proceso hijo.

**NOTA:** Si pretende utilizar el formato numérico para el ID de usuario y/o de grupo, necesita introducir un símbolo # antes del valor numérico, que puede encontrar en los archivos /etc/passwd y /etc/group.

## ServerAdmin

ServerAdmin define la dirección de correo electrónico que se muestra cuando el servidor genera un error en la página. Asignelo a su dirección de correo electrónico.

A continuación tiene que asignar el nombre del host al servidor utilizando la directiva servername. Esta directiva se comenta por defecto porque la instalación de Apache no puede adivinar qué nombre de host utilizar en su sistema. Por lo tanto, si el host se llama www.domain.com, fije la directiva ServerName de forma correspondiente.

**NOTA:** Asegúrese, sin embargo, de que ese nombre de host que está introduciendo tiene registrados los nombres de dominio de servidor adecuados para que lo dirija hacia la máquina en la que se encuentra su servidor.

## DocumentRoot

Al igual que ocurre con otros servidores Web, Apache necesita conocer la ruta del directorio de máximo nivel en el que se guardarán las páginas Web. Este directorio se denomina normalmente directorio raíz de documentos. Apache proporciona una directiva llamada DocumentRoot, que se puede utilizar para especificar la ruta de este directorio.

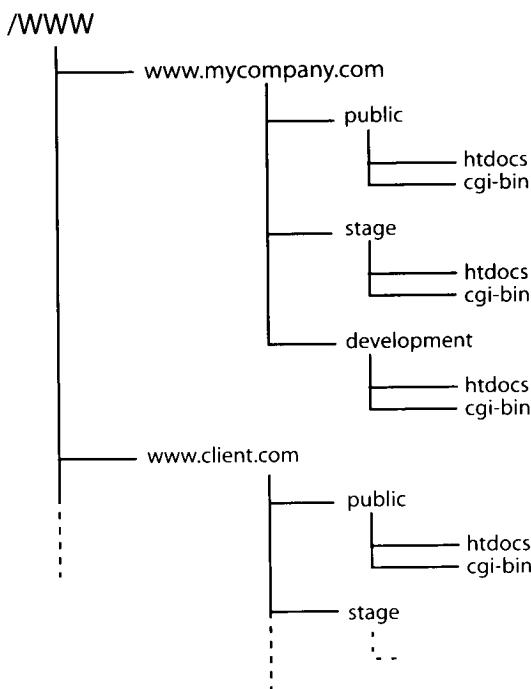
Esta directiva instruye al servidor para que maneje el directorio suministrado como el directorio raíz para todos los documentos. Ha de tomar una decisión muy importante. Por ejemplo, si la directiva se fija en:

```
DocumentRoot /
```

entonces cada archivo del sistema se vuelve accesible para el servidor Web. Por supuesto, puede proteger archivos proporcionando asignación adecuada de permisos a archivos, pero asignar la raíz del documento al directorio raíz físico de su sistema es definitivamente un riesgo de seguridad mayor. En vez de esto, podría dirigir la raíz de documentos a un subdirectorio específico en su sistema de archivos. Si ha utilizado la opción --prefix=/usr/local/apache en la configuración de la fuente de Apache, esta directiva será:

```
DocumentRoot "/usr/local/apache/htdocs"
```

Una opción potencialmente mejor, sin embargo, podría ser crear una estructura de directorios Web para su organización. La figura 3.2 nos muestra la estructura de directorios Web que aconsejo para un sistema multiusuario y multidominio.



**Figura 3.2.** La estructura de directorios Web que aconsejo

Tal y como muestra la figura, he elegido crear una partición llamada /www, y bajo ella hay directorios para cada sitio Web alojado en mi sistema. /www/www.mycompany.com/ tiene tres subdirectorios: public, stage y development. Cada uno de estos subdirectorios tiene, a su vez, dos subdirectorios: htdocs y cgi-bin. El subdirectorio htdocs es el directorio raíz del documento, y el subdirectorio cgi-bin se utiliza para scripts CGI. Por lo tanto, el DocumentRoot asignado para el sitio Web www.mycompany.com es:

```
DocumentRoot "/www/www.mycompany.com/public/htdocs"
```

La ventaja de esta estructura de directorios es que mantiene todos los documentos Web y todas las aplicaciones bajo una partición (/www). Esto permite backups sencillos, y la partición se puede montar en distintos sistemas mediante el Network File System (NFS), en caso de que se le asigne la tarea de proporcionar páginas Web a otra máquina en la red. Discutiremos el diseño de estructuras de directorios Web en mayor profundidad más adelante en este libro.

Tenga en cuenta que el que su raíz de documentos esté en un directorio determinado, no significa que el servidor Web no pueda acceder a directorios fuera del

árbol de documentos. Puede permitirlo fácilmente utilizando enlaces simbólicos (con permisos de archivo apropiados) o utilizando alias (que se discuten en el siguiente capítulo).

**ADVERTENCIA:** Para una perspectiva organizada y segura, no recomiendo la utilización de muchos enlaces simbólicos o alias para acceder a archivos y directorios fuera de su árbol de documentos. Sin embargo, en ocasiones es necesario mantener cierto tipo de información fuera del árbol de documentos, incluso si necesita mantener el contenido de tal directorio accesible para el servidor en una base uniforme. Si tiene que añadir enlaces simbólicos a otras localizaciones de directorios fuera del árbol de documentos, asegúrese de que cuando copie sus archivos, su programa de backup pueda copiar también los enlaces simbólicos adecuadamente.

## Directivas en contenedores de directorios

El siguiente conjunto de directivas está encerrado en un contenedor <Directory . . .> tal y como se muestra a continuación:

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
```

El alcance de las directivas encerradas está limitado a dicho directorio (con cualquier subdirectorío); sin embargo, únicamente deberá utilizar las directivas que están permitidas en un contexto de directorios (aprenderá más sobre estas directivas en el siguiente capítulo).

Aquí, las directivas Options y AllowOverride se aplican al documento %DocumentRoot% al que está enraizado (/) o al directorio de máximo nivel del sitio Web principal. Como las directivas están encerradas en contenedores de directorios se aplican a todos los subdirectorios pertenecientes al directorio nombrado, las directivas se aplican a todos los directorios dentro de %DocumentRoot%.

La directiva Options está asignada a FollowSymLinks, que le dice a Apache que le permita a sí misma atravesar cualquier simbólico dentro de %DocumentRoot%. Como la directiva Options está asignada sólo a los siguientes enlaces simbólicos, no hay otras opciones disponibles para ningún directorio dentro de %DocumentRoot%. Efectivamente, la directiva Options es:

```
Options FollowSymLinks -ExecCGI -Includes -Indexes -MultiViews
```

Las otras opciones están explicadas en la sección de la directiva Options del siguiente capítulo. Sin embargo, tenga en cuenta que la gran idea aquí es crear un servidor muy cerrado. Como sólo están permitidos los enlaces simbólicos trans-

versales, debe permitir explícitamente otras opciones necesarias en un directorio. Se trata de un buen arreglo para una seguridad eventual. El siguiente contenedor de directorios abre el directorio %DocumentRoot% del modo siguiente:

```
<Directory "/usr/local/apache/htdocs">
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

Si su %DocumentRoot% es distinto, cambie la ruta de dicho directorio. A continuación se explica el significado de la configuración anterior para Apache:

- El directorio y sus subdirectorios se pueden indexar. Si existe un archivo con el índice, se mostrará; en ausencia de este archivo, el servidor creará un índice dinámico para el directorio. Esto se especifica con la directiva Options.
- Tanto el directorio como todos sus subdirectorios pueden tener enlaces simbólicos que el servidor puede seguir (es decir, utilizarlos como ruta) para acceder a información. La directiva Options también se ocupa de esto.
- El directorio y todos sus subdirectorios pueden tomar parte en negociaciones de contenido. La opción MultiViews para la directiva Options determina esta posibilidad. No soy muy partidario de esta opción pero no me desagrada hasta el punto de llegar a eliminarla. Por ejemplo, cuando la directiva Options está permitida dentro del directorio %DocumentRoot% como se muestra arriba, se puede responder a una solicitud `http://www.domain.com/ratecard.html` con un archivo llamado `ratecard.html.bak`, `ratecard.bak` o `ratecard.old`, y lo mismo ocurre si `ratecard.html` está ausente. Esto puede ser deseable o puede no serlo.
- Ninguna de las opciones especificadas pueden ser ignoradas por un archivo de control de acceso local (especificado por la directiva AccessFileName en `httpd.conf`; el valor por defecto es `.htaccess`). Esto se especifica utilizando la directiva AllowOverride.
- Las directivas Allow se evalúan antes de las directivas Deny. El acceso se deniega por defecto. Cualquier cliente que no corresponda a una directiva Allow o que corresponda a una directiva Deny tiene denegado el acceso al servidor.
- El acceso está permitido para todos.

La asignación por defecto debería ser suficiente.

**ADVERTENCIA:** Si su servidor va a estar en Internet, es posible que quiera eliminar la opción `FollowSymlinks` para la línea de la directiva `Options`. Dejar esta opción crea un riesgo potencial de seguridad. Por ejemplo, si un directorio en su sitio Web no tiene una página con un índice, el servidor despliega un índice automático que muestra cualquier enlace simbólico que tenga en ese directorio. Esto puede causar el despliegue de información reservada, o quizás incluso, permitir que cualquiera ejecute un ejecutable que resida en un directorio mal enlazado.

## UserDir

La directiva `UserDir` le dice a Apache que considere `%UserDir%` como una raíz de documentos (`~username/%UserDir%`) de cada usuario del sitio Web. Esto sólo tiene sentido si dispone de varios usuarios en el sistema y quiere permitir que cada uno de ellos tenga su propio directorio Web. La asignación por defecto es:

```
UserDir public_html
```

que significa que si decide que el nombre de su servidor Web sea `www.yourcompany.com`, y tiene dos usuarios (Joe y Jenny), las URL de sus páginas Web personales serían:

```
http://www.yourcompany.com/~joe      Physical directory: ~joe/
public_html
http://www.yourcompany.com/~jenny    Physical directory: ~jenny/
public_html
```

Tenga en cuenta que en los sistemas Unix, `~` (tilde) se aplica también al directorio local del usuario. El directorio especificado por la directiva `UserDir` reside en el directorio local de cada usuario, y Apache ha de leer y ejecutar permisos para leer archivos y directorios dentro del directorio `public_html`. Esto se puede realizar utilizando el siguiente comando en un sistema Unix:

```
chown -R <user>.<Apache server's group name>
~<user>/<directory assigned in UserDir>
chmod -R 770 ~<user>/<directory assigned in UserDir>
```

Por ejemplo, si el nombre de usuario es `joe` y el grupo de Apache es `httpd`, y asignamos `public_html` en la directiva `UserDir`, el comando anterior se convierte en el siguiente:

```
chown -R joe.httpd ~joe/public_html
chmod -R 2770 ~joe/public_html
```

El primer comando, `chown`, cambia el propietario del directorio `~joe/public_html` (y ocurre lo mismo con todos los archivos y subdirectorios dentro de él) a `joe.httpd`. En otras palabras, le da al usuario `joe` y al grupo

`httpd` la propiedad absoluta de todos los archivos y directorios en el directorio `public_html`. El siguiente comando, `chmod`, fija el acceso justo en 2770, en otras palabras, únicamente el usuario (`joe`) y el grupo (`httpd`) tienen privilegios absolutos de lectura, escritura y de ejecución en `public_html` y en todos sus archivos y subdirectorios.

Además asegura que cuando se crea un nuevo archivo o subdirectorio en el directorio `public_html`, el nuevo archivo creado tiene asignado el ID de grupo. Esto permite que el servidor Web acceda al nuevo archivo sin la intervención del usuario.

**TRUCO:** Si crea una cuenta de usuario en su sistema utilizando un script (como el script `/usr/sbin/adduser` en un sistema Linux), posiblemente quiera incorporar el proceso de creación del sitio Web en este script. Para crear un directorio `public_html` por defecto, simplemente añada un comando `mkdir` (si eso es lo que ha asignado a la directiva `UserDir`). Añada los comandos `chmod` y `chown` para dar al servidor Web permiso de usuario para leer y ejecutar archivos y directorios bajo este directorio público.

## DirectoryIndex

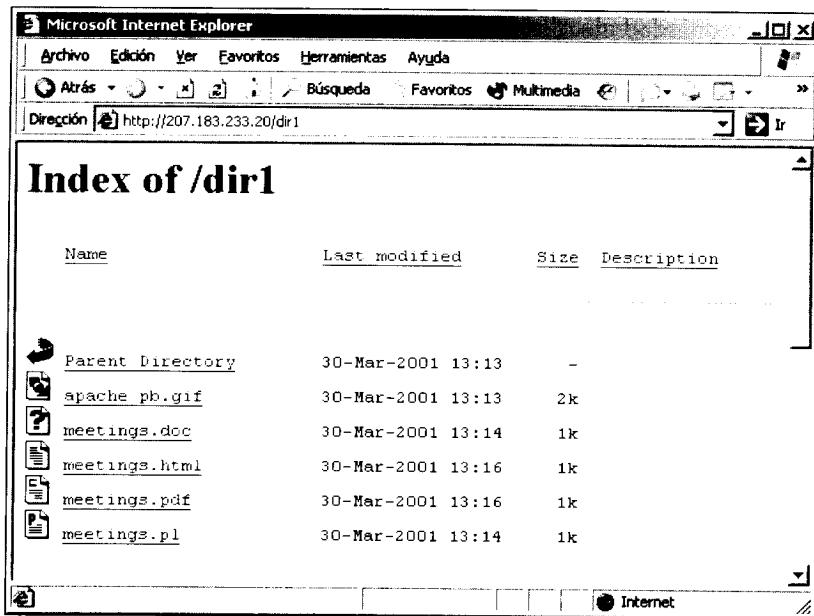
A continuación, necesita configurar la directiva `DirectoryIndex`, que tiene la siguiente sintaxis:

```
DirectoryIndex [filename1, filename2, filename3, ... ]
```

Esta directiva determina qué archivo del servidor Apache se podría considerar como el índice para el directorio que se ha solicitado. Por ejemplo, cuando se solicita una URL como `www.yourcompany.com/`, el servidor Apache determina que esta es una solicitud para acceder al directorio `/` (raíz del documento) del sitio Web. Si se asigna la directiva `DocumentRoot` del siguiente modo:

```
DocumentRoot "/www/www.yourcompany.com/public/htdocs"
```

entonces el servidor Apache busca un archivo llamado `/www/www.yourcompany.com/public/htdocs/index.html`; si encuentra el archivo, Apache sirve la solicitud devolviendo el contenido del archivo al navegador Web que lo ha solicitado. Si se asigna el `DirectoryIndex` a `welcome.html` en lugar de a `default index.html`, por el contrario, el servidor Web buscará `/www/www.yourcompany.com/public/htdocs/welcome.html`. Si el archivo está ausente, Apache devuelve la lista de directorios creando una página HTML. La figura 3.3 muestra lo que ocurre cuando `index.html` está ausente en un directorio y el servidor ha generado un listado de directorios para el navegador que lo ha solicitado.



**Figura 3.3.** Listado dinámico de directorios en ausencia de index.htm

Puede especificar varios nombres de archivo para el índice en la directiva DirectoryIndex. Por ejemplo:

```
DirectoryIndex index.html index.htm welcome.htm
```

le dice al servidor Web que debería revisar la existencia de cualquiera de los tres archivos, y si encuentra uno de los archivos, será devuelto al cliente Web que lo solicitó.

**NOTA:** Hacer una lista de tantos archivos como tenga el índice puede crear dos problemas. Primero, el servidor tendrá que verificar la existencia de muchos archivos para las solicitudes del directorio; esto podría hacerlo más lento de lo habitual. Segundo, tener varios archivos como índices, podría hacer su sitio difícil de manejar desde el punto de vista de la organización. Si los desarrolladores del contenido de su sitio Web utilizan varios sistemas para crear archivos, podría ser una solución práctica guardar ambos archivos, index.html e index.htm, como archivos de índice. Por ejemplo, una máquina antigua de Windows es incapaz de crear nombres de archivos con extensiones mayores de tres caracteres, por lo que un usuario trabajando en esa máquina necesitará actualizar manualmente todos los archivos index.htm de usuario en el servidor Web. Utilizar los nombres de archivo recomendados elimina este problema.

## **AccessFileName**

La directiva AccessFileName define el nombre del archivo de configuración de control de acceso a cada directorio. El nombre por defecto .htaccess tiene un período destacado para ocultar el archivo de la lista de directorios normal bajo los sistemas Unix. La única razón para cambiar el nombre es aumentar la seguridad por medio de la ocultación. Si cambia el nombre de archivo, asegúrese de que cambia la expresión "`^\.ht`" a "`^\.whatever`" donde `.whatever` es el primer carácter visible de lo que asigna a AccessFileName.

## **Contenedor de archivos**

El siguiente contenedor `<Files . . .>` le dice a Apache que no permita el acceso a ningún archivo que comience con un ht (es decir, a el .htaccess o al .htpasswd.) Esto corresponde a `%AccessFileName%`.

```
<Files ~ "^\.\.ht">
    Order allow,deny
    Deny from all
</Files>
```

## **UseCanonicalName**

La siguiente directiva es UseCanonicalName, que está fijada en On. Le dice a Apache que cree todas las URL de auto referencia utilizando el formato `%ServerName% : %Port%`. Dejarlo así es una buena idea.

## **TypesConfig**

La directiva TypesConfig pertenece al archivo de configuración mime mime.types que reside en el directorio por defecto conf. No necesita cambiarlo aunque tenga este archivo en una localización distinta.

## **DefaultType**

La directiva DefaultType asigna la cabecera Content-Type a cualquier archivo cuya MIME no pueda determinarse desde la extensión del archivo. Por ejemplo, si tiene un archivo `%DocumentRoot%/myfile`, entonces Apache utiliza el `%DefaultType`, que está fijado en `text/plain`, como el tipo de contenido para el archivo. Esto significa que cuando el navegador Web solicita y recibe ese archivo como respuesta, desplegará el contenido en el mismo sentido que muestra un archivo todo texto. Si piensa que la mayoría del contenido de los archivos desconocidos debería tratarse como HTML, entonces utilice `text/html` en lugar de `text/plain`.

## **Contenedor IfModule**

El siguiente contenedor `<IfModule . . .>` le dice a Apache que permita el módulo mágico MIME (`mod_mime_magic`) si éste existe, y que utilice el

`%MIMEMagicFile%` como la información mágica (patrones de bytes) necesaria para identificar archivos de tipo MIME.

Debería dejar el valor por defecto como está, a no ser que quiera cambiar la ruta del archivo mágico.

```
<IfModule mod_mime_magic.c>
    MIMEMagicFile conf/magic
</IfModule>
```

## HostnameLookups

La directiva `HostnameLookups` le dice a Apache que permita la búsqueda de DNS para cada solicitud si ésta se ha fijado en `On`. Sin embargo, el valor por defecto es `Off` y por lo tanto no tiene lugar la búsqueda de DNS para procesar la solicitud, lo que alargaría el tiempo de respuesta. Realizar una búsqueda de DNS para resolver una dirección IP para el nombre del host, es un paso que consume tiempo en un servidor ocupado y sólo debería realizarse utilizando la utilidad `logresolve` tal y como se discute en el capítulo 8. Deje el valor por defecto como está.

## ErrorLog

La directiva `ErrorLog` es muy importante. Señala el archivo de registro dedicado al registro de errores del servidor. El valor por defecto de registros/errores traduce a `%ServerRoot%/logs/error_log`, que trabajará para usted, a no ser que quiera escribir un registro en un sitio distinto. Generalmente, es una buena idea crear una partición de registro para guardar sus registros. Además, es preferible que su partición de registros esté en uno o más discos dedicados de registro. Si tiene ese tipo de configuración de hardware, querrá cambiar la directiva para determinar una nueva ruta de registro.

## LogLevel

La directiva `LogLevel` asigna el nivel de registro que llevaremos a cabo. El valor por defecto `warn` es suficiente para empezar.

Las directivas `LogFormat` dictan lo que está registrado y en qué formato está registrado. En la mayor parte de los casos, será capaz de trabajar con los valores por defecto. Por lo tanto no deberíamos hacer ningún cambio hasta que leamos el capítulo 8.

## CustomLog

La directiva `CustomLog` asigna la ruta para el registro de acceso que almacena los eventos de su servidor. Por defecto utiliza el formato de registro común (CFL), que se ha definido en la directiva `LogFormat`. Considere el siguiente consejo y guarde los registros en su propio disco y partición, y realice los cambios en la ruta si es necesario.

**TRUCO:** Un buen consejo para todos los registros, independientemente del directorio en el que guarde los registros, es asegurarse de que el único que tiene acceso de escritura en ese directorio es el proceso principal del servidor. Esta es la mayor medida de seguridad, porque permitir a otros usuarios o procesos escribir en el directorio de registros puede significar que alguien sin autorización sea capaz de hacerse con el poder de su proceso UID principal de su servidor Web, que habitualmente es la cuenta raíz.

## ServerSignature

La siguiente directiva es `ServerSignature`, que despliega el nombre del servidor y el número de versión y es una página generada por el servidor como las páginas dinámicas de índices de directorios, las páginas de error, así como este tipo de páginas. Si se siente incómodo con la posibilidad de desplegar información sobre su servidor, fije el valor en `Off`. Yo lo hago.

## Alias

La directiva `Alias` define un nuevo directorio de alias llamado `/icons/` que se encuentra en `/usr/local/apache/icons/` (es decir, `%ServerRoot%/icons/`). Las imágenes de iconos, almacenadas en este directorio, se utilizan para desplegar listas dinámicas de directorios cuando no se encuentran archivos especificados por `%DirectoryIndex%` en ese directorio. Debe dejar el alias como está a no ser que cambie la ruta del directorio de iconos. El contenedor del directorio que sigue la definición de alias asigna el permiso para este directorio de iconos. No me gusta la idea de que esté permitida la navegación por los directorios (es decir, indexado dinámico de directorios) fijando `Options` en `Indexes`. Debe cambiar `Options Indexes` a `Options -Indexes` y no preocuparse por la opción `MultiViews`.

## ScriptAlias

La directiva `ScriptAlias` se utiliza para asignar un directorio de alias de scripts CGI muy utilizado, el directorio `/cgi-bin/` que se encuentra en `/usr/local/apache/cgi-bin/` (es decir, `%ServerRoot%/cgi-bin/`). Si tiene pensado utilizar scripts CGI en el servidor principal, mantenga esta directiva; en caso contrario, elimínela. Por otro lado, si quiere cambiar el directorio de scripts a otra localización, cambie la ruta física dada en la directiva para que coincida con la suya.

**ADVERTENCIA:** Nunca asigne la ruta CGI a un directorio dentro de su raíz de documentos, es decir, `%DocumentRoot% /somepath`, porque mantener scripts CGI en su directorio de la raíz de documentos le enfrenta

**a varias cuestiones de seguridad. Asigne su ruta de scripts CGI y el DocumentRoot al mismo nivel. En otras palabras, si fija DocumentRoot en /a/b/c/htdocs, entonces coloque ScriptAlias en /a/b/c/cgi-bin y no en /a/b/c/htdocs/cgi-bin o en /a/b/c/htdocs/d.cgi-bin.**

A continuación, un contenedor de directorios marca una restricción en el directorio %ScriptAlias% para asegurar que no se permitan las opciones de nivel de directorio.

Aquí la directiva Options se fija en None, que significa que el contenido de %ScriptAlias% no es navegable, ese enlace simbólico dentro del directorio %ScriptAlias% no se puede seguir.

## El resto de directivas

El resto de las directivas: IndexOptions; AddIconByEncoding; AddIconByType; AddIcon; DefaultIcon; ReadmeName; HeaderName; IndexIgnore; AddEncoding; AddLanguage; AddCharset; BrowserMatch; y AddType, no son importantes para la preparación y la ejecución, por lo que las ignoraremos por ahora. Podrá ver estas directivas en el capítulo 4.

Sin embargo, hay dos directivas que puede tener la necesidad de cambiar: LanguagePriority y AddDefaultCharset.

### LanguagePriority

La directiva LanguagePriority está fijada con el valor en (inglés), que será el idioma por defecto, lo que no tiene por qué servirle a todo el mundo. Por lo tanto, puede cambiarlo a su idioma materno, si éste se soporta.

### AddDefaultCharset

AddDefaultCharset asignará el conjunto de caracteres que mejor se ajuste a sus necesidades locales. Si no sabe qué conjunto de caracteres fijar, puede dejar el valor por defecto, determinar qué conjunto de caracteres debería utilizar, y cambiar el valor por defecto más tarde.

## Iniciar y parar Apache

Una vez que ha personalizado httpd.conf, está listo para iniciar el servidor. En esta sección, asumo que ha tenido en cuenta mi consejo (es decir, colocar --prefix en /usr/local/apache) del capítulo anterior. Si no ha seguido mi consejo, entonces asegúrese de que cambia todas las referencias a /usr/local/apache a la referencia adecuada.

# Iniciar Apache

Ejecute el comando `/usr/local/apache/bin/apachectl start` para iniciar el servidor Web Apache. Si apachectl tiene quejas sobre errores de sintaxis, deberá corregir los errores en el archivo `httpd.conf` y reintentarlo.

Además ha de comprobar si hay algún mensaje de error en el archivo de registro `%ErrorLog%` (es decir, `/usr/local/apache/logs/error_log`). Si observa errores en el archivo de registro, tendrá que corregirlos antes. Los errores más comunes son:

- **No ejecutar el servidor como el usuario raíz.** Debe iniciar Apache como usuario raíz. Una vez que Apache se ha iniciado, producirá procesos hijo que utilizarán las directivas `User` y `Group`. La mayor parte de la gente tiene problemas con este punto e intenta iniciar el servidor utilizando la cuenta de usuario especificada en la directiva `User`.
- **Apache parece incapaz de "atarse" a una dirección.** Esto ocurre tanto si otro proceso está listo para utilizar el puerto que ha configurado Apache, como si está ejecutando `httpd` como un usuario normal pero tratando de utilizar un puerto por debajo de 1024 (como el puerto por defecto, que es el 80).
- **Perder la ruta de los archivos de configuración.** Asegúrese de que existen las rutas `%ErrorLog%` y `%CustomLog%` y que sólo puede escribirlas el servidor Apache.
- **Erratas de configuración.** Cada vez que cambie el archivo de configuración, ejecute `/usr/local/apache/apachectl configtest` para comprobar que no tiene un error de sintaxis en el archivo de configuración.

**TRUCO:** La forma más rápida de comprobar si el servidor está funcionando es con el siguiente comando:

```
ps auxw | grep httpd
```

Este comando utiliza la utilidad `ps` para realizar una lista de todos los procesos que se encuentran en la cola de procesos, y entonces conduce esta salida hacia el programa `grep`. `grep` busca la salida en líneas que coincidan con la palabra clave `httpd`, y entonces muestra cada línea con coincidencias. Si observa una línea con la palabra `root`, ese es su proceso principal del servidor Apache. Tenga en cuenta que cuando se inicia el servidor, crea un número de procesos hijo para manejar las solicitudes. Si inicia Apache como el usuario raíz, el proceso padre continua ejecutando como `root`, mientras que los hijos cambian al usuario siguiendo las instrucciones del archivo `httpd.conf`. Si está ejecutando Apache en Linux, puede crear el script que se muestra en el listado 3.2 y guardararlo en

el directorio /etc/rc.d/init.d/. Este script le permite iniciar y parar Apache de forma automática cuando está reiniciando el sistema.

### Listado 3.2. El script httpd

```
#!/bin/sh
#
# httpd  Este shell script inicia y para el servidor Apache
# Toma un argumento 'start' o 'stop' para iniciar o parar
# el proceso del servidor.
#
# Notas: Tendría que cambiar la información de la ruta
# utilizada en el script para reflejar la configuración de
# su sistema
#
APACHECTL=/usr/local/apache/bin/apachectl

[ -f $APACHECTL ] || exit 0

# Ver cómo se llamaba el script
case "$1" in
    start)
        # Iniciar demonios .
        echo -n "Starting httpd: "
        $APACHECTL start
        touch /var/lock/subsys/httpd
        echo
        ;;
    stop)
        # Stop daemons.
        echo -n "Shutting down httpd: "
        $APACHECTL stop

        echo "done"
        rm -f /var/lock/subsys/httpd
        ;;
    *)
        echo "Usage: httpd {start|stop}"
        exit 1
esac
exit 0
```

**TRUCO:** Para iniciar Apache de forma automática cuando arranca su sistema, simplemente ejecute este comando una vez:

```
ln -s /etc/rc.d/init.d/httpd /etc/rc.d/rc3.d/S99httpd
```

Este comando crea un enlace especial llamado S99httpd en el directorio /etc/rc.d/rc3.d (nivel de ejecución 3) que enlaza al script /etc/

**rc.d/init.d/httpd. Cuando su sistema arranca, este script será ejecutado con el argumento start y Apache se iniciará automáticamente.**

## Reiniciar Apache

Para reiniciar el servidor Apache ejecute el comando `/usr/local/apache/bin/apachectl restart`.

También puede utilizar el comando `kill` del siguiente modo:

```
kill -HUP `cat /usr/local/apache/logs/httpd.pid`
```

Cuando reiniciamos con `apachectl restart` o utilizando la señal HUP con `kill`, el proceso padre de Apache (se ejecuta como usuario raíz) mata todos sus hijos, lee el archivo de configuración y reinicia una nueva generación de hijos a medida que los necesita.

**NOTA:** Esta forma de reiniciar es inesperada para los clientes Web a los que se les prometió que serían servidos por los procesos hijo que estaban vivos en ese momento. Por tanto, debería considerar utilizar `graceful` con `apachectl` en lugar de la opción `restart`, y `WINCH` en lugar de la señal HUP con el comando `kill`. En ambos casos, el proceso padre de Apache aconsejará a sus procesos hijo finalizar la solicitud actual y terminar para que pueda volver a leer el archivo de configuración y reiniciar un nuevo lote de hijos. Esto gastaría tiempo en un sitio con mucho tráfico.

## Parar Apache

Puede parar Apache de forma automática cuando se reinicia el sistema, o de forma manual en cualquier momento. Estos dos métodos de parar Apache se discuten en las siguientes secciones.

### Parar Apache automáticamente

Para finalizar Apache automáticamente cuando se está reiniciando el sistema, ejecute este comando una vez:

```
ln -s /etc/rc.d/init.d/httpd /etc/rc.d/rc3.d/K99httpd
```

Este comando asegura que el script `httpd` se está ejecutando con el argumento `stop` cuando el sistema se apaga.

### Parar el servidor Apache manualmente

Para parar el servidor Apache, ejecute el comando `/usr/local/apache/bin/apachectl stop`.

El servidor Apache también lo utiliza para encontrar el PID en la raíz de procesos del servidor Web. El PID está escrito en un archivo asignado a la directiva `PidFile`. Este es el PID para el proceso `httpd` principal del servidor Web. No intente matar los procesos hijo manualmente uno a uno, porque el proceso padre los recreará a medida que los necesite. Otro modo de parar el servidor Apache es ejecutar:

```
kill -TERM `cat /usr/local/apache/logs/httpd.pid`
```

Este comando ejecuta el comando `kill` con la señal `-TERM` (es decir, `-9`) para el proceso ID que ha devuelto el comando `cat /usr/local/apache/logs/httpd.pid` (es decir, `cat %PidFile%`).

## Comprobar Apache

Una vez que ha iniciado el servidor Apache, puede acceder a él mediante un navegador Web utilizando el nombre de host adecuado. Por ejemplo, si está ejecutando el navegador Web desde el servidor, entonces utilice `http://localhost/` para acceder. Sin embargo, si quiere acceder al servidor desde un host remoto, utilice el nombre del host completo para el servidor. Por ejemplo, para acceder a un servidor llamado `apache.pcnltd.com`, utilice `http://apache.pcnltd.com`. Si asigna la directiva `Port` a un puerto no estándar (es decir, distinto de 80), entonces recuerde incluir el `port` en la URL. Por ejemplo, `http://localhost:8080` accederá al servidor Apache en el Puerto 8080. Si no ha realizado ningún cambio en el directorio por defecto `htdocs`, verá una página como la que se muestra en la figura 3.4. Esta página está construida con la distribución Apache y tiene que reemplazarla con su propio contenido. Para terminar, ha de asegurar los archivos de registro y actualizarlos adecuadamente. Para comprobar sus archivos de registro, introduzca el directorio `log` y ejecute el siguiente comando:

```
tail -f path_to_access_log
```

La parte `tail` del comando es una utilidad Unix que permite ver un archivo en crecimiento (cuando se especifica la opción `-f`). Asegúrese de que cambia `path_to_access_log` a un nombre de ruta completo adecuado para el registro de acceso. Ahora, utilice un navegador Web para acceder al sitio; si ya está en el sitio, simplemente actualice la página que tiene en el navegador. Debería ver una entrada añadida a la lista en la pantalla. Presione el botón de actualizar unas cuantas veces más para asegurar que el archivo de acceso está actualizado. Si ve los registros adecuados, su archivo de registro está funcionando. Presione `Ctrl+C` para salir de la sesión del comando `tail`. Si no ve ningún registro en el archivo, debe verificar las asignaciones de permisos para los archivos de registro y el directorio en el que están guardados.

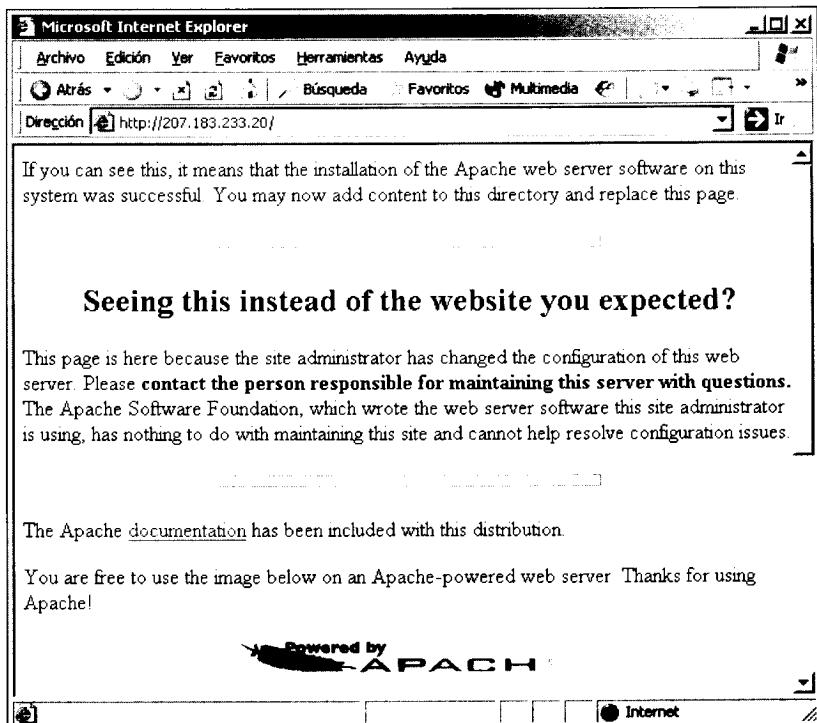


Figura 3.4. Página de inicio por defecto en Apache

Otro registro que es necesario comprobar es el archivo de registro de errores. Utilice:

```
tail -f path_to_error_log
```

para ver las entradas de registro de errores tal y como entran. La simple petición de recursos inexistentes (como un archivo que no usted tiene) para ver en su navegador Web, hará que se vean las entradas que se han añadido. Si observa entradas que se han añadido, entonces el archivo de registro de errores está bien configurado.

Si todas estas pruebas han tenido éxito, entonces ha configurado con éxito su servidor Apache. ¡Felicitaciones!



# 4 Configurar Apache con directivas MPM Winnt

---

## En este capítulo

1. Entendemos los contextos de las directivas Apache.
2. Nos familiarizamos con las directivas básicas.
3. Configuramos Apache con las directivas MPM threaded.
4. Configuramos Apache con las directivas MPM prefork.
5. Configuramos Apache con las directivas MPM perchild.

Una directiva es simplemente un comando al que obedece Apache. Apache lee directivas de los archivos de configuración que hemos discutido en el capítulo 3. Utilizando directivas, un administrador de Apache puede controlar el comportamiento del servidor Web. Apache tiene muchas directivas disponibles, que le convierten en un servidor Web enormemente configurable. Las directivas que forman parte de la instalación básica de Apache se llaman directivas básicas. Éstas están siempre disponibles. Los módulos estándar tienen disponibles muchas otras directivas desde que forman parte de la distribución estándar de Apache. Esas directivas estándar basadas en módulos, se discuten a lo largo de este libro.

Este capítulo discute los contextos estándar en los que se aplican las directivas, proporcionando una información completa sobre las directivas básicas, y también sobre las directivas disponibles en distintos módulos multiproceso (o MPM) que se han introducido en Apache 2.0. En lugar de realizar una lista por orden alfabético de todas las directivas básicas posibles, las he agrupado de acuerdo con su uso; las categorías incluyen configuración general; configuración de rendimiento y de recursos; contenedores estándar; host virtuales específicos; registro; y autenticación y seguridad. En cada descripción de una directiva se proporciona la siguiente información:

**Sintaxis:** muestra el nombre de la directiva y todos sus argumentos posibles o valores que toma.

**Predefinido:** esta línea muestra el valor por defecto para una directiva. Sólo se muestra cuando se puede aplicar.

**Contexto:** especifica el contexto (o alcance) en el que se aplica la directiva.

**Invalidar:** valor necesario para permitir la directiva en el archivo de las directivas de configuración de acceso en el ámbito de directorio (.htaccess por defecto). Sólo se muestra donde se puede aplicar.

Lo primero que vamos a ver en este capítulo son los contextos en los que puede utilizar estas directivas.

**NOTA:** Algunas directivas aparecen en varias listas; tan sólo una de estas listas contiene la discusión completa de estas directivas, y se puede encontrar en cualquier parte del capítulo. Esto ocurre porque algunas directivas no se ajustan a una sola categoría, y quiero que vea estas directivas desde sus distintas perspectivas.

## Contextos de las directivas Apache

Antes de que utilice alguna de las directivas básicas, es importante que entienda en qué contexto se utiliza la directiva; en otras palabras, necesita conocer el contexto (o alcance) de las directivas. Una vez discutida la terminología necesaria para las directivas básicas, las describiré.

Hay tres contextos principales en los que se utilizan las directivas:

- **Contexto de configuración del servidor:** las directivas pueden aparecer en cualquier archivo de configuración del servidor principal, fuera de los contenedores (que se parecen mucho a las etiquetas HTML).

- **Contexto de contenedor:** las directivas se encuentran en contenedores del siguiente tipo:

```
<Container_name>  
  Directivas  
</Container_name>.
```

- **Contexto de nivel de directorio:** las directivas están almacenadas en un archivo (habitualmente en .htaccess) dentro de un directorio.

## Contexto de configuración del servidor

Las directivas pueden aparecer en cualquier archivo de configuración del servidor principal, fuera de los contenedores. Puede pensar en este contexto como el contexto general o de alcance general; es decir, manejan directivas que no están encerradas en contenedores como en el caso de las *directivas generales*. Las directivas que se aplican en este contexto afectan por defecto al resto de los contextos. Estas directivas se podrían utilizar en cualquier archivo de configuración del servidor (como httpd.conf, srm.conf y access.conf), pero no dentro de un contenedor o en un archivo de configuración en el ámbito de directorios (.htaccess).

## Contexto de contenedor

Para limitar el alcance de las directivas, puede utilizar contenedores, que son como las etiquetas HTML. Un par de etiquetas encierra un conjunto de directivas, restringiendo el alcance de las mismas. Apache ofrece los siguientes contenedores estándar:

- <VirtualHost ...> ... </VirtualHost> se utiliza para aplicar una o más directivas al host virtual que se especifica en la etiqueta abierta del contenedor.
- <Directory ...> ... </Directory> se utiliza para aplicar una o más directivas a un directorio determinado. Tenga en cuenta que si especifica una o más directivas para un directorio utilizando esta etiqueta de contenedor, las directivas se aplican automáticamente a todos sus subdirectorios. Si no desea este efecto secundario, puede crear un contenedor de directorio separado para cada subdirectorio y controlar el comportamiento del servidor de forma distinta en cada subnivel del directorio.
- <DirectoryMatch regex> ... <DirectoryMatch> es exactamente el mismo que el contenedor <Directory>; sin embargo, toma una expresión regular (regex) como argumento, en lugar del nombre de un directorio. Por ejemplo, <DirectoryMatch "^\www/mydir [1-

3] / "> ... </DirectoryMatch> corresponde a todos los directorios llamados /www/mydir1, /www/mydir2 y /www/mydir3.

Una expresión regular (regex) está compuesta normalmente por caracteres normales y especiales para crear un patrón. Este patrón se utiliza para buscar coincidencias con una o más subcadenas o con una cadena completa. Ver el apéndice B para obtener más información sobre expresiones regulares.

- <Files ...> ... </Files> se utiliza para aplicar una o más directivas a un archivo determinado o a un grupo de archivos.
- <FilesMatch regex> ... </FilesMatch> es exactamente igual que el contenedor <Files>; sin embargo toma una expresión regular (regex) como argumento en lugar de uno o más nombres de archivo. Por ejemplo, <FilesMatch "\.(doc|txt)\$"> ... </FilesMatch> aplicará una o más directivas a todos los archivos que terminen con las extensiones .doc o .txt.
- <Location ...> ... </Location> se utiliza para aplicar una o más directivas a una URL determinada.

**NOTA:** URI (Uniform Resource Identifier) es el término genérico para la familia de Uniform Resource Identifiers, siendo uno de sus miembros URL. El resto de los miembros son Uniform Resource Names (URN), Uniform Resource Characteristics (URC) y Location-Independent File Names (LIFN). URL es la que se utiliza realmente.

- <LocationMatch regex> ... </LocationMatch> es exactamente igual que el contenedor <Location>; sin embargo, toma una expresión regular (regex) como argumento en lugar de una URI.
- <Limit ...> ... </Limit> se utiliza para aplicar una o más directivas al control de acceso a determinadas áreas de un sitio Web o a un método de solicitud HTTP determinado. Este contenedor tiene el menor alcance de todos los contenedores. A continuación tenemos un ejemplo del alcance de un contenedor: un segmento de un archivo httpd.conf:

```
<VirtualHost 206.171.50.50>
ServerName www.nitec.com
DocumentRoot "/www/nitec/public/htdocs"
DirectoryIndex welcome.html
<Location /secured/>
    DirectoryIndex login.html
</Location>
</VirtualHost>
```

En este ejemplo, un host virtual llamado www.nitec.com se define utilizando el contenedor <VirtualHost>. Las tres directivas ServerName,

`DocumentRoot` y `DirectoryIndex`, están en contexto del host virtual, y por lo tanto se aplican al host virtual completo. Las directivas `DirectoryIndex` determinan que si se realiza una solicitud para acceder a un directorio en este servidor virtual, se devolverá un archivo llamado `welcome.html` si este se encuentra disponible. Sin embargo, el contenedor `<Location>` determina que se devolverá un archivo distinto, `login.html`, cuando alguien intente acceder a la URL `www.nitec.com/secured/`. Como el contenedor `<Location>` define un alcance menor (en el subdirectorio `/secured`), invalida el mayor alcance de la directiva `DirectoryIndex` en el contenedor `<VirtualHost>`.

**NOTA:** Un contenedor que define un menor alcance siempre invalida al contenedor de mayor alcance.

Debe recordar algunas reglas cuando utilice alguno de los contenedores para definir el comportamiento de una sección en su espacio Web:

- Un contenedor `<VirtualHost>` no se puede anidar dentro de otro contenedor sea del tipo que sea.
- No puede haber un contenedor dentro del contenedor de contexto de menor alcance, `<Limit>`.
- Un contenedor `<Files>` únicamente puede tener el contenedor de menor alcance dentro de sí mismo, `<Limit>`.
- Los contenedores `<Location>` y `<Directory>` no se mezclan, por lo tanto, no utilice uno dentro del otro.

## Contexto en el ámbito de directorio

También puede incluir directivas en los archivos de configuración en el ámbito de directorio. Un archivo de configuración en el ámbito de directorio (el nombre de archivo por defecto para la configuración en el ámbito de directorio es `.htaccess`) es un archivo de texto que contiene una o más directivas que se aplican únicamente en el archivo actual. Estas directivas se pueden encerrar también en contenedores como `<Files ...>` o `<Limit ...>`. Utilizando los archivos de configuración en el ámbito de directorio, puede controlar el comportamiento de Apache cuando se realiza una solicitud de un archivo en un directorio.

**NOTA:** Las directivas `AllowOverride` le permiten invalidar todo o parte de lo que se puede invalidar en un archivo de configuración en el ámbito de directorio en el contexto del servidor o del host virtual. Por lo tanto, es posible que no se puedan procesar todas las directivas en este contexto, dependiendo de la invalidación que esté activada.

# Directivas de configuración general

Las directivas discutidas en esta sección están fundamentalmente en experimentación y por norma general se aplican tanto al servidor principal (contexto de configuración del servidor) como a los servidores virtuales (contexto de host virtual).

## AccessFileName

Las directivas AccessFileName determinan el nombre del archivo de control de acceso en el ámbito de directorio. La asignación por defecto (.htaccess) hace que Apache busque el archivo .htaccess cada vez que un sistema cliente realiza una solicitud de acceso.

**Sintaxis:** AccessFileName nombre de archivo [nombre de archivo ...]

**Predefinido:** AccessFileName .htaccess

**Contexto:** configuración del servidor, host virtual

Por ejemplo, imagine que las directivas DocumentRoot de un poderoso sitio Apache llamado www.mycompany.com tienen como asignación DocumentRoot "/www/mycompany/public/htdocs" y un navegador Web solicita la página http://www.mycompany.com/feedback.html. Esto hace que Apache busque los siguientes archivos de control de acceso:

- /.htaccess
- /www/.htaccess
- /www/mycompany/.htaccess
- /www/mycompany/public/.htaccess
- /www/mycompany/public/htdocs/.htaccess

Hasta que Apache no ha verificado todos estos archivos no empieza a buscar el archivo feedback.html. Si piensa que todo esto implica demasiadas salidas y entradas, es cierto. Puede evitarlo especificando el nombre del archivo de acceso en estas directivas.

**TRUCO:** Si no utiliza el archivo de control de acceso en el Ámbito de directorios y le gustaría que Apache dejase de verificarlo, simplemente utilice las directivas <Directory> para invalidar los privilegios de invalidación de opciones, del siguiente modo:

```
<Directory />
    AllowOverride None
</Directory>
```

Ver las secciones de contenedores `<Directory>` y las directivas `AllowOverride` de este capítulo para obtener más detalles.

## AddDefaultCharset

Las directivas `AddDefaultCharset` asignan el carácter por defecto a la cabecera `Content-Type` enviada por Apache al navegador.

**Sintaxis:** `AddDefaultCharset On | Off | juego de caracteres`

**Predefinido:** `AddDefaultCharset Off`

**Contexto:** todos

Cuando estas directivas se activan (utilizando la opción `On`), Apache envía `iso-8859-1` (`Western European`) como el carácter por defecto a no ser que especifiquemos un carácter como la segunda opción para esta directiva. Por ejemplo, `AddDefaultCharset On utf-8` enviará `UTF-8` como carácter por defecto asignado. Las asignaciones de caracteres más comunes son:

- ISO-8859-1 — Western European
- ISO-8859-15 — Western European with Euro currency symbol support
- Windows-1252 — Western European
- CP850 — Western European
- UTF-8 — 8-bit Unicode
- UTF-7 — 7-bit Unicode

**NOTA:** Si sus documentos HTML asignan el carácter utilizando la etiqueta `<META http-equiv="Content-Type" content="content_type; charset=character_set_name">`, las directivas `AddDefaultCharset` le permiten invalidarlas.

## ContentDigest

Cuando se fijan las directivas `ContentDigest`, se genera una cabecera de la digestión de mensajes con un algoritmo MD5 para todo el contenido del cuerpo,

que permite que el cliente Web verifique la integridad de la página. Este es el principal arrastre de rendimiento para el servidor Web porque la digestión MD5 tiene que computerizarse en cada página estática servida por el servidor Web.

Tenga en cuenta que esta digestión no se genera por ninguna salida generada por ningún módulo distinto del básico. Esto significa que la salida Common Gateway Interface (CGI) no puede utilizar esta característica. Dado que se trata de un sumidero de rendimiento en el servidor, no se recomienda esta característica a no ser que sepa que su servidor tiene poder de sobra.

**Sintaxis:** ContentDigest On | Off

**Predefinido:** ContentDigest Off

**Contexto:** todos

## DefaultType

Las directivas DefaultType se utilizan para establecer un tipo de contenido por defecto, de modo que cuando Apache recibe una solicitud de un documento cuyo tipo de archivo es desconocido (en otras palabras, no se puede determinar desde la asociación de tipos MIME disponibles para el servidor), utiliza el tipo MIME predeterminado.

**Sintaxis:** DefaultType mime-type

**Predefinido:** DefaultType text/html

**Contexto:** todos

**Invalidar:** FileInfo

Por ejemplo, si tiene un directorio en el que guarda muchos archivos de texto sin extensiones, puede utilizar las directivas DefaultType dentro de un contenedor `<Directory>` que se encuentre en este contenedor. En este caso, la asignación de DefaultType en text/plain le permite al servidor decirle a la otra parte (el navegador Web) que esos son archivos todo texto. A continuación tenemos un ejemplo:

```
<Directory /www/mycompany/public/htdocs/plaindata>
DefaultType plain/text
</Directory>
```

Aquí, todos los archivos del directorio `/www/mycompany/public/htdocs/plaindata/` se tratan como todo texto.

## DocumentRoot

El directorio DocumentRoot especificado por esta directiva se convierte en el directorio de máximo nivel para todos los documentos servidos por Apache.

**Sintaxis:** DocumentRoot "ruta del directorio"

**Predefinido:** DocumentRoot "/usr/local/apache/htdocs"

**Contexto:** configuración del servidor, host virtual

Por ejemplo, si se asigna:

```
DocumentRoot "/www/mycompany/public/htdocs"
```

al servidor www.mycompany.com, entonces una solicitud de acceso a www.mycompany.com/corporate.html hace que el servidor busque el siguiente archivo:

```
/www/mycompany/public/htdocs/corporate.html.
```

Si encuentra el archivo, lo devuelve al cliente (es decir, al navegador Web).

**NOTA:** Un error en el módulo mod\_dir da lugar a un problema cuando el DocumentRoot tiene una barra final (por ejemplo, DocumentRoot /usr/web/), por lo que debe evitar introducir un carácter / al final de la ruta de cualquier directiva.

**TRUCO:** Es posible que el servidor busque archivos en un directorio fuera del directorio DocumentRoot. Si quiere acceder a algunos archivos fuera del árbol DocumentRoot, puede utilizar las directivas Alias para crear un nombre de directorio virtual que pertenezca a un directorio físico en alguna localización del sistema de archivos de su servidor.

## ErrorDocument

Cuando el servidor encuentra un problema, genera un mensaje de error estándar con el código de error en él. Su utilización no es demasiado intuitiva para la mayoría de las personas, por el contrario, sería más deseable una versión de mensajes de error más personalizada o, posiblemente, una solución nueva. Si necesita tal personalización, utilice las directivas ErrorDocument para invalidar los mensajes de error estándares.

**Sintaxis:** ErrorDocument error\_code [nombre de archivo | mensaje de error | URL]

**Predefinido:** ninguno

**Contexto:** todos

**Invalidar:** FileInfo

Estas directivas necesitan dos argumentos. El primer argumento es el código de error estándar HTTP, que puede encontrar en el apéndice A; el segundo argumento es la acción para el error. Dependiendo de sus necesidades, puede definir qué acción quiere que tome el servidor para una condición determinada de error.

Por ejemplo, si quiere proporcionar un mensaje personalizado para todas las solicitudes que den lugar a un mensaje estándar del tipo "file not found" ("no encuentra el archivo"), tendrá que encontrar el código del estado del servidor para esa condición de error y utilizar las directivas ErrorDocument. Dado que el código del estado del servidor para los archivos desaparecidos es 404, la asignación de las directivas siguientes le permiten a Apache mostrar un mensaje personalizado:

```
ErrorDocument 404 "Lo siento, esta no es una solicitud válida  
porque %s "
```

Tenga en cuenta que todo el mensaje se encuentra entre comillas, y el servidor reemplaza %s con aquella información respecto al error que encuentre disponible. Si encuentra todo esto un poco limitado, puede utilizar un archivo como mensaje de error. Por ejemplo:

```
ErrorDocument 404 /errors/404.html
```

Cada vez que tenga lugar un error por pérdida de un archivo, se devolverá al cliente (el navegador Web) el archivo 404.html que encontramos en el directorio de errores bajo el directorio DocumentRoot.

Si necesita más control sobre el manejo de errores, puede devolver un mensaje personalizado que incluya un script CGI para llevar a cabo algunas acciones específicas. En tal caso, reemplace el nombre del archivo con una llamada a un script CGI:

```
ErrorDocument 404 /cgi-bin/missingurl.cgi
```

Este script llama a un script CGI llamado missingurl.cgi cada vez que tiene lugar un error 404. Además puede redirigir al cliente a otro sitio autorizando una URL en lugar del nombre de archivo:

```
ErrorDocument 404 http://www.newsite.com/wemoved.html
```

Este comando utilizar cuando se ha cambiado la localización de una página o de un directorio.

**NOTA:** No puede dirigir el cliente a un servidor remoto si tiene lugar un error 401 (no autorizado). El valor de estas directivas debe ser un archivo local o un recurso.

## <IfDefine>

La directiva del contenedor `IfDefine` le permite crear una configuración condicional. Se especifica la opción `special_command_line_param` utilizando la opción `-D` con el programa `httpd`.

**Sintaxis:** `<IfDefine [!]special_command_line_param> ... </IfDefine>`

**Predefinido:** ninguno

**Contexto:** todos

Por ejemplo, si ejecuta el servidor Apache desde el directorio bin como `./httpd -D something`, entonces puede utilizar:

```
<IfDefine something>
    # #directivas que deberian ejecutarse sólo cuando
    # -D something está especificado
</IfDefine>
```

**NOTA:** Colocando un carácter `!` frente a un `special_command_line_param`, se permite definir a Apache que no debe ejecutar las directivas dentro del contenedor `IfDefine` únicamente cuando no se especifique el `-D something` en el command-line. Por ejemplo:

```
<IfDefine !something>
    # directivas que deberian ejecutarse sólo cuando
    # NO está -D something especificado
</IfDefine>
```

## <IfModule>

Utilice las directivas del contenedor `IfModule` si tiene directivas que están disponibles desde un módulo personalizado que no está siempre presente en su instalación Apache.

**Sintaxis:** `<IfModule [!]module_name> ... </IfModule>`

**Predefinido:** ninguno

**Contexto:** todos

Por ejemplo, si quiere utilizar determinadas directivas únicamente si un módulo está disponible, entonces puede utilizar la siguiente construcción condicional:

```
<IfModule module_name>
    # Asigna las a siguientes directivas sus valores correspondientes
    # si el módulo forma parte de Apache.
```

```
# Sus directivas van aquí.  
</IfModule>
```

cuando el argumento `module_name` es el nombre de archivo del módulo en el momento en que se compiló (por ejemplo, `mod_rewrite.c`).

Si necesita una sentencia condicional que sea exactamente la contraria de esta última, lo único que necesita hacer es insertar un ! (signo de exclamación) antes del nombre del módulo. Las secciones `<IfModule>` son anidables; este método se puede utilizar para implementar pruebas de condición sencillas con varios módulos. Por ejemplo:

```
<IfModule module_A>  
    # Procesa las directivas aquí si el módulo A forma parte de  
    # Apache  
        <IfModule module_B>  
            # Se coloca aquí únicamente si el módulo A y el B  
            # forman parte de Apache  
                <IfModule ! module_C>  
                    # Se coloca aquí únicamente si el módulo A y el B  
                    # pero no el C, existen como parte de Apache  
                </IfModule>  
        </IfModule>  
    </IfModule>  
</IfModule>
```

## Include

Las directivas `Include` le permiten incluir un archivo externo como un archivo de configuración.

**Sintaxis:** `Include nombre_de_archivo`

**Predefinido:** ninguno

**Contexto:** configuración del servidor

Por ejemplo, si quiere bajar todas las configuraciones de su host virtual utilizando archivos externos, puede tener la siguiente configuración en `httpd.conf`:

```
NameVirtualHost IP_Address  
Include virtual_host_1.conf  
Include virtual_host_2.conf  
Include virtual_host_3.conf  
...  
Include virtual_host_N.conf
```

**TRUCO:** En cada uno de estos archivos puede definir un contenedor `<VirtualHost>` específico para el host. Se trata de una buena forma de organizar el archivo `httpd.conf` en el caso de que tenga muchos host virtuales.

# Options

Las directivas Options controlan qué características del servidor están disponibles en un directorio determinado.

**Sintaxis:** Options [+|-]opción [+|-]opción ...

**Predefinido:** ninguno

**Contexto:** todos

**Invalidar:** Option (ver tabla 4.1)

**NOTA:** Cuando esta directiva se fija en None, no están disponibles ninguna de las características extra para el contexto en el que se utilizan las directivas.

La tabla 4.1 muestra todas las asignaciones posibles para estas directivas.

**Tabla 4.1.** Asignaciones en las directivas Options

Asignación	Significado
None	No hay opciones.
All	Todas las opciones excepto para MultiViews.
ExecCGI	Está permitida la ejecución de scripts CGI.
FollowSymLinks	El servidor sigue los enlaces simbólicos en el directorio. Sin embargo, el servidor no cambia el nombre de ruta utilizado para enfrentarse a las secciones <Directory>.
Includes	Están permitidos los comandos SSI.
IncludesNOEXEC	Se puede embeber un conjunto restringido de comandos SSI en páginas SSI. Los comandos SSI que no están disponibles son #exec y #include.
Indexes	Si se solicita una URL que está integrada en un directorio y no hay DirectoryIndex (por ejemplo, index.html) en ese directorio, entonces el servidor devuelve una lista formateada del directorio.
SymLinksIfOwnerMatch	El servidor únicamente sigue enlaces simbólicos en los casos en los que el archivo o el directorio objetivo es propiedad del mismo usuario que el enlace.

Asignación	Significado
MultiViews	Permite negociación de contenido basado en un lenguaje de documentos.

Utilice los signos + y - para activar o desactivar una opción en la directiva Options. Por ejemplo, el siguiente segmento de configuración muestra dos contenidores de directorios en un solo archivo de configuración como el access.conf:

```
<Directory /www/myclient/public/htdocs >
    Options Indexes MultiViews
</Directory>

<Directory /www/myclient/public/htdocs>
    Options Includes
</Directory>
```

El /www/myclient/public/htdocs únicamente tendrá el conjunto de opciones Includes. Sin embargo, si la segunda sección de <Directory> utiliza los signos + y - del siguiente modo:

```
<Directory /www/myclient/public/htdocs>
    Options +Includes -Indexes
</Directory>
```

las opciones MultiViews e Includes se asignan al directorio especificado. Cuando aplique varias opciones Options, tenga cuidado y recuerde que el menor contexto prevalece sobre el mayor. Por ejemplo:

```
ServerName www.domain.com
Options ExecCGI Includes
<VirtualHost 11.22.33.11.22.33.1>
    ServerName www.myclient.com
    Options -ExecCGI -Includes
    <Directory /www/myclient/public/htdocs/ssi >
        Options Includes
    </Directory>
</VirtualHost>
```

En este ejemplo, el servidor principal permite tanto la ejecución de CGI como la de SSI asignando la directiva Options a ExecCGI y a Includes. Sin embargo, el host virtual www.myclient.com no permite ninguna de estas dos opciones, pues utiliza las asignaciones -ExecCGI y -Includes en su propia directiva Options. Por último, el host virtual tiene otra directiva Options para el directorio /www/myclient/public/htdocs/ssi, que permite la ejecución de SSI. Observe que Includes es la única opción que está asignada al directorio /www/myclient/public/htdocs/ssi.

Como puede ver, si la directiva Options utiliza los signos + o -, los valores se añaden o se restan de la lista actual de Options. Por otro lado, si la directiva Options no utiliza los signos relativos + o -, los valores para ese contenedor, las directivas Options anteriores serán totalmente invalidadas para ese contenedor.

## Port

Las directivas Port asignan a un host un número de puerto en el rango de 0 a 65535. En ausencia de cualquier directiva Listen o BindAddress que especifique un número de puerto, la directiva Port asigna el puerto de red al servidor que va a escuchar. Si alguna directiva Listen o BindAddress especifica un número de puerto, entonces la directiva Port no tiene efecto sobre la elección de la dirección a la que escucha el servidor. La directiva Port asigna la variable de entorno SERVER\_PORT (para CGI y Server-Side Include (SSI)), y se utiliza cuando el servidor debe generar una URL que se refiera a él mismo.

**Sintaxis:** Port número

**Predefinido:** Port 80

**Contexto:** configuración del servidor

Aunque puede especificar un número de puerto entre 0 y 65535, hay una restricción que debería recordar. Todos los puertos por debajo de 1024 están reservados para servicios estándares como TELNET, SMTP, POP3, HTTP y FTP. Puede localizar todas las asignaciones a servicios estándar en el archivo /etc/services. O si quiere estar seguro, utilizar cualquier número de puerto distinto de 80 para su servidor Apache (utilice una dirección alta, como 8000, por ejemplo).

**NOTA:** Si no es un usuario raíz y quiere ejecutar Apache para practicar o por alguna otra causa, necesita utilizar puertos superiores a 1024, porque únicamente un usuario raíz puede ejecutar servicios como Apache en estos puertos reservados.

**TRUCO:** El contenedor <VirtualHost> se puede utilizar también para determinar qué puerto se utiliza en un host virtual.

## ServerAdmin

La directiva ServerAdmin asigna una dirección de correo electrónico que aparece junto con muchos mensajes de error en el servidor. Si aloja una gran

cantidad de sitios Web virtuales, es posible que quiera utilizar distintas direcciones de correo electrónico para cada host virtual y así, de ese modo, poder determinar inmediatamente de qué servidor está hablando el informe de problemas.

**Sintaxis:** ServerAdmin e-mail

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

Para darle un aspecto profesional a sus sitios virtuales y para que este aspecto sea merecido, no utilice una dirección de correo electrónico que no incluya el sitio virtual como la parte host de la dirección.

Por ejemplo, si su compañía es un proveedor de servicios de Internet (ISP) llamado `mycompany.net`, y tiene un sitio cliente llamado `www.myclient.com`, entonces asigne `www.myclient.com` ServerAdmin a la dirección `user@myclient.com` como `webmaster@myclient.com`, en lugar de `webmaster@mycompany.net`. De este modo, cuando el servidor muestre un mensaje de error a algún visitante de `www.myclient.com`, el visitante verá una dirección de correo electrónico que pertenece a `myclient.com`. esto se considera mucho más profesional.

## ServerName

La directiva `ServerName` asigna el nombre del host del servidor. Cuando no se utiliza esta directiva, Apache intenta determinar el nombre del host haciendo que un servidor de nombres de dominio (DNS) lo solicite en la puesta en marcha. Dependiendo de su sistema DNS, sin embargo, puede que esto no sea deseable, porque la búsqueda realizada por Apache puede elegir un nombre para su servidor indeseable, por ejemplo, en el caso de que tenga un registro canónico de nombres (CNAME) para su servidor.

Por lo tanto, lo mejor es simplemente asignarlo al nombre de host que prefiere.

**Sintaxis:** ServerName fully\_qualified\_domain\_name

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

**TRUCO:** Asegúrese de que introduce el nombre de dominio completo en lugar de solo una parte. Por ejemplo, si tiene un host llamado `wormhole.mycompany.com`, no debe asignar el `ServerName` en `wormhole`. La elección correcta es:

`ServerName wormhole.mycompany.com`

## **ServerRoot**

La directiva `ServerRoot` asigna el directorio en el que residen los archivos del servidor. No lo confunda con la directiva `DocumentRoot`, la cual se utiliza para dirigir al servidor a sus contenidos Web. La directiva `ServerRoot` se utiliza para localizar todos los archivos de configuración del servidor y los archivos de registro. La distribución estándar incluye los directorios `conf`, `bin`, `htdocs`, `icons`, `cgi-bin` y `logs` bajo el directorio `ServerRoot`. Si no especifica la directiva `ServerRoot`, puede utilizar la opción `-d command-line` para decirle a Apache cuál es su directorio `ServerRoot`.

**Sintaxis:** `ServerRoot directorio`

**Predefinido:** `ServerRoot /usr/local/apache`

**Contexto:** configuración del servidor

## **ServerSignature**

Utilizando la directiva `ServerSignature` puede crear un solo pie de página para que Apache genere páginas como los mensajes de error y las listas de directorios. No se recomienda esta directiva a no ser que utilice Apache como servidor proxy. Por otro lado, cuando un usuario recibe un mensaje de error, normalmente es difícil determinar qué servidor proxy está causando el error si hay una cadena de proxies en la ruta de red del usuario. Este pie de página actúa como un identificador en este tipo de casos. Puede incluir una dirección de correo electrónico que aparecerá en el pie de página para que se pongan en contacto con usted vía correo electrónico si hubiese algún problema.

**Sintaxis:** `ServerSignature On | Off | e-mail`

**Predefinido:** `ServerSignature Off`

**Contexto:** todos

## **ServerTokens**

Apache puede enviar una cabecera que incluya un identificador que le diga al cliente qué servidor está ejecutando, en respuesta a una solicitud. La directiva `ServerTokens` le permite controlar esa señal identificadora. Cuando se utiliza la opción `Minimal`, Apache envía "Apache/version"; cuando se utiliza la opción `ProductOnly`, sólo se envía la cadena "Apache"; cuando se utiliza `OS` se envía "Apache/version (OS\_Type)"; cuando se utiliza `Full`, Apache envía "Apache/version (OS\_Type) Available\_Module\_Info".

**Sintaxis:** `ServerTokens Minimal | ProductOnly | OS | Full`

**Predefinido:** ServerTokens Full

**Contexto:** configuración del servidor

**ADVERTENCIA:** Recomiendo utilizar únicamente la opción Minimal si quiere evitar problemas de seguridad de ataques basados en error; obviamente no quiere que todo el mundo sepa el tipo de servidor que está utilizando.

## SetInputFilter

La directiva SetInputFilter asigna los filtros que utilizaremos para procesar una solicitud enviada al servidor. Estos filtros se aplican en el orden en el que aparecen en esta directiva.

**Sintaxis:** SetInputFilter filtro [filtro ...]

**Predefinido:** ninguno

**Contexto:** directorio

## SetOutputFilter

La directiva SetOutputFilter asigna los filtros que utilizaremos para procesar una respuesta antes de que se envíe al cliente Web. Estos filtros se aplican en el orden en el que aparecen en esta directiva.

**Sintaxis:** SetOutputFilter filtro [filtro] ...

**Predefinido:** ninguno

**Contexto:** directorio

En el siguiente ejemplo, todos los archivos en el directorio /www/mysite/htdocs/parsed se procesarán utilizando el filtro de salida INCLUDES, que es el filtro SSI:

```
<Directory "/www/mysite/htdocs/parsed">
    Options +Includes
    SetOutputFilter INCLUDES
</Directory>
```

## Directivas de rendimiento y de configuración de recursos

Estas directivas le permiten ajustar Apache para llegar a un rendimiento más alto y a un mejor control. Puede ajustar los procesos Apache de varias formas.

Tenga en cuenta que la mayor parte de estas directivas necesitan que comprenda realmente el modo en el que trabaja su sistema en términos de sistema operativo, hardware, y similares; por tanto, debería consultar el manual de su sistema operativo para aprender la forma en la que limita a los procesos los recursos del sistema, cómo controla la conexión TCP/IP, y este tipo de eventos. Las directivas en esta sección están divididas en subfunciones.

Más adelante en este libro, obtendremos más información sobre cómo aumentar la velocidad de Apache.

## Controlar los procesos de Apache

Las siguientes directivas se utilizan para controlar la ejecución de Apache en su sistema. La utilización de estas directivas le permite controlar el modo en el que Apache utiliza los recursos en su sistema. Por ejemplo, puede decidir cuántos procesos hijo se van a ejecutar en su sistema, o cuántos hilos podría permitir que utilice Apache en una plataforma Windows. Es necesario recordar los siguientes puntos a la hora de configurar estas directivas:

- Cuantos más procesos ejecute, mayor carga experimenta la CPU.
- Cuantos más procesos ejecute, más memoria RAM necesita.
- Cuantos más procesos ejecute, más recursos del sistema operativo (como descriptores de archivos y buffers compartidos) se utilizan.

Por supuesto, más procesos también pueden significar más solicitudes servidas y, por lo tanto, más éxitos para su sitio Web. Por este motivo, la asignación de estas directivas debe basarse en una combinación de experimentación, requisitos y recursos disponibles.

### **ListenBacklog**

Ver la directiva `ListenBacklog` en la sección "Directivas específicas de MPM threaded".

### **MaxClients**

Ver la directiva `MaxClients` en la sección "Directivas específicas de MPM threaded".

### **MaxRequestsPerChild**

Ver la directiva `MaxRequestsPerChild` en la sección "Directivas específicas de MPM threaded".

### **MaxSpareServers**

Ver la directiva `MaxSpareServers` en la sección "Directivas específicas de MPM prefork".

## **MinSpareServers**

Ver la directiva `MinSpareServers` en la sección "Directivas específicas de MPM prefork".

## **SendBufferSize**

Ver la directiva `SendBufferSize` en la sección "Directivas específicas de MPM threaded".

## **StartServers**

Ver la directiva `StartServers` en la sección "Directivas específicas de MPM threaded".

## **TimeOut**

El servidor Apache responde a las solicitudes. Las solicitudes y las respuestas se transmiten mediante paquetes de datos. Apache debe saber el tiempo que ha de esperar un paquete determinado. La directiva `TimeOut` le permite configurar el tiempo en segundos. El tiempo que se especifica aquí es el tiempo máximo que Apache esperará antes de romper la conexión. La asignación por defecto permite que Apache espere 300 segundos antes de desconectarse del cliente. Si se encuentra en una red lenta, sin embargo, es posible que prefiera aumentar ese valor para disminuir el número de desconexiones.

**Sintaxis:** `TimeOut number`

**Predefinido:** `TimeOut 300`

**Contexto:** configuración del servidor

Esta asignación `TimeOut` se aplica a:

- El tiempo total que tarda en recibir una solicitud GET.
- El tiempo entre la recepción de paquetes TCP en una solicitud POST o en una solicitud PUT.
- El tiempo entre las respuestas ACKs de las transmisiones de los paquetes TCP.

## **Realizar conexiones persistentes**

Utilizando las directivas `KeepAlive` discutidas en esta sección, puede dar instrucciones a Apache para que utilice conexiones persistentes y, de ese modo, se pueda utilizar una sola conexión TCP para varias transacciones. Normalmente, cada solicitud HTTP y cada respuesta utilizan una conexión separada. Esto significa que cada vez que el servidor obtiene una solicitud, abre una conexión para recuperar la solicitud y luego la cierra. Una vez que el servidor ha recibido la

solicitud, abre otra conexión TCP para responder y, finalmente, cierra la conexión una vez que ha completado el servicio. Este método disminuye el rendimiento. Reutilizar una sola conexión para varias transacciones reduce la carga necesaria para establecer y cerrar una conexión TCP de forma repetida, y por lo tanto aumenta el rendimiento.

Para establecer una conexión permanente, sin embargo, tanto el servidor como el cliente necesitan tener la posibilidad de una conexión permanente. La mayor parte de los navegadores populares, como Netscape Navigator y Microsoft Internet Explorer, tienen las características KeepAlive incorporadas.

No todas las transacciones pueden sacar partido a las conexiones permanentes. Un requisito para una conexión permanente es que los recursos que se transmiten tengan un tamaño conocido. Como muchos scripts CGI, comandos SSI, y otros contenidos generados dinámicamente no tienen un tamaño conocido antes de la trasmisión, no son capaces de sacar partido de esta característica.

## KeepAlive

La directiva KeepAlive le permite activar / desactivar la utilización persistente de conexiones TCP en Apache.

**Sintaxis:** KeepAlive On | Off

**Predefinido:** KeepAlive On

**Contexto:** configuración del servidor

**NOTA:** Los antiguos servidores Apache (versiones anteriores a la versión 1.2) necesitaban un valor numérico en lugar de On/Off cuando utilizaban KeepAlive. Este valor corresponde al número máximo de solicitudes que quiere que Apache tome en consideración por solicitud. Se impone un límite para prevenir que el cliente acapare todos los recursos del servidor. Para invalidar KeepAlive en las versiones antiguas de Apache, utilice el 0 (cero) como valor.

## KeepAliveTimeout

Si tiene la directiva KeepAlive asignada con el valor on, puede utilizar la directiva KeepAliveTimeout para limitar el número de segundos que Apache debe esperar una solicitud posterior antes de cerrar la conexión. Una vez que se recibe la solicitud, se aplica el tiempo de espera especificado en la directiva Timeout.

**Sintaxis:** KeepAliveTimeout segundos

**Predefinido:** KeepAliveTimeout 15

**Contexto:** configuración del servidor

## **MaxKeepAliveRequests**

La directiva MaxKeepAliveRequests limita el número de solicitudes permitidas por conexión cuando KeepAlive tiene asignado el valor `on`. Si tiene asignado el valor 0 (cero), se permitirán ilimitadas solicitudes. Recomiendo que se asigne un valor alto para conseguir un máximo rendimiento del servidor.

**Sintaxis:** MaxKeepAliveRequests `número`

**Predefinido:** MaxKeepAliveRequests 100

**Contexto:** configuración del servidor

## **Controlar los recursos del sistema**

Apache es muy flexible a la hora de permitirle controlar la cantidad de recursos del sistema (como el tiempo CPU y la memoria) que consume. Estas características de control son convenientes para aumentar la fiabilidad y la sensibilidad del sistema de su servidor Web.

Muchos intentos de ataques se basan en conseguir que un servidor Web consuma todos los recursos del sistema como si fuera un sumidero y, por tanto, intentan que el sistema se vuelva insensible y se pare virtualmente. Apache aporta un conjunto de directivas para combatir este tipo de situaciones. Estas directivas se discuten en las siguientes secciones.

### **RLimitCPU**

La directiva RLimitCPU le permite controlar la utilización que hace la CPU de los procesos hijo producidos por Apache como son los scripts CGI. El límite no se aplica a los procesos hijo de Apache ni a ningún proceso creado por servidor padre de Apache.

**Sintaxis:** RLimitCPU `n | 'max' [ n | 'max' ]`

**Predefinido:** no asignado; utiliza los valores por defecto del sistema operativo

**Contexto:** configuración del servidor, host virtual

La directiva RLimitCPU toma los dos parámetros siguientes: el primer parámetro asigna un límite de recursos bajo para todos los procesos y el segundo parámetro, que es opcional, asigna el límite máximo de recursos. Tenga en cuenta que esa elevación del límite máximo de recursos requiere que se ejecute el servidor como raíz o en la fase inicial del arranque. Para cada uno de estos parámetros hay dos valores posibles:

- `n` es el número de segundos por proceso.
- `y max` es el límite máximo de recursos permitido por el sistema operativo.

## **RLimitMEM**

La directiva RLimitMEM limita la utilización de memoria (RAM) de los procesos hijo producidos por Apache como los scripts CGI. El límite no se aplica a los hijos Apache ni a ningún proceso creado por el servidor padre Apache.

**Sintaxis:** RLimitMEM n | 'max' [ n | 'max' ]

**Predefinido:** no asignado; utiliza los valores por defecto del sistema operativo

**Contexto:** configuración del servidor, host virtual

La directiva RLimitMEM toma dos parámetros. El primer parámetro asigna un límite bajo para todos los procesos y el segundo parámetro, que es opcional, asigna el límite máximo de recursos.

Tenga en cuenta que ese aumento en el límite de recursos requiere que el usuario raíz arranque el servidor arranque. Hay dos valores posibles para cada uno de estos parámetros:

- n es el número de bytes por proceso
- max es el límite máximo de procesos permitido por el sistema operativo

## **RLimitNPROC**

La directiva RLimitNPROC asigna el máximo número de procesos hijo generados por Apache por ID de usuario.

**Sintaxis:** RLimitNPROC n | 'max' [ n | 'max' ]

**Predefinido:** no asignado; utiliza los valores por defecto del sistema operativo

**Contexto:** configuración del servidor, host virtual

La directiva RLimitNPROC toma dos parámetros. El primer parámetro asigna un límite bajo para todos los procesos y el segundo parámetro, que es opcional, asigna el límite máximo de recursos. La elevación del límite máximo de recursos requiere que se ejecute el servidor como raíz o en la fase inicial del arranque. Hay dos valores posibles para cada uno de estos parámetros:

- n es el número de bytes por proceso
- max es el límite máximo de procesos permitido por el sistema operativo

**NOTA:** Si se están ejecutando sus procesos CGI bajo el mismo ID de usuario al igual que el proceso del servidor, la utilización de RLimitNPROC limita el número de procesos que el servidor puede lanzar (o "forkear"). Si el límite es muy bajo, recibirá un mensaje del tipo "Cannot fork process".

**("no se puede bifurcar el proceso") en el archivo de registro de errores. En ese caso, deberá incrementar el límite o simplemente dejarlo en su valor por defecto.**

## UseCanonicalName

La directiva `UseCanonicalName` determina el modo en el que Apache construye las URL referidas a sí mismas. Cuando se le asigna el valor `on`, Apache utiliza las asignaciones de las directivas `ServerName` y `Port` para crear la URL referida a sí misma. Si `UseCanonicalName` tiene `off` como valor asignado, entonces Apache utiliza el nombre de host suministrado por el cliente y el número de puerto de la información de cabecera para construir la URL. Por último, si `UseCanonicalName` tiene asignado el valor `dns`, Apache llevará a cabo una búsqueda DNS inversa en la dirección IP del servidor para determinar el nombre del host para la URL. No se recomienda esta opción porque la búsqueda DNS inversa ralentizará el proceso de solicitudes.

**Sintaxis:** `UseCanonicalName On | Off | dns`

**Predefinido:** `UseCanonicalName On`

**Contexto:** configuración del servidor, host virtual, directorio

**Invalidar:** opciones

## Utilizar módulos dinámicos

Apache carga todos los módulos precompilados cuando se inicia; sin embargo, también proporciona una característica de carga y descarga dinámica de módulos que nos puede resultar útil en ciertas ocasiones. Cuando utilice las siguientes directivas de módulos dinámicos, puede cambiar la lista de módulos activos sin recompilar el servidor.

### AddModule

La directiva `AddModule` se puede utilizar para permitir un módulo precompilado que actualmente no está activo. El servidor puede tener módulos compilados que no se están utilizando en este momento. Esta directiva se puede utilizar para permitir estos módulos. El servidor contiene una lista precargada de módulos activos; esta lista puede borrarse con la directiva `ClearModuleList`. Entonces se pueden añadir módulos utilizando la directiva `AddModule`.

**Sintaxis:** `AddModule módulo módulo ...`

**Predefinido:** ninguno

**Contexto:** configuración del servidor

## **ClearModuleList**

Puede utilizar la directiva `ClearModuleList` para borrar la lista de módulos activos y permitir la característica de carga de módulos dinámicos. Entonces utilice la directiva `AddModule` para añadir los módulos que quiera activar.

**Sintaxis:** `ClearModuleList`

**Predefinido:** Ninguno

**Contexto:** configuración del servidor

## **Directivas de contenedores estándar**

Esta sección discute los contenedores estándar que forman parte del servidor base de Apache. Estos contenedores son muy utilizados en la aplicación de un grupo de otras directivas a determinados directorios, archivos o localizaciones. No puede mezclar y emparejar de forma aleatoria estos contenedores.

Las líneas generales de trabajo al utilizar estas directivas son:

- Utilizar los contenedores `<Directory>` o `<Files>` para especificar directivas para los objetos de sistemas de archivos como son los archivos y directorios. No puede utilizar `<Directory>` dentro de un archivo `.htaccess`, porque un archivo `.htaccess` se aplica únicamente al directorio en el se ha encontrado.
- Utilizar el contenedor `<Location>` para emparejar objetos URL. No puede utilizar esta directiva dentro de un archivo `.htaccess`.
- Cuando utilice una versión de una expresión regular de una directiva (por ejemplo, `<DirectoryMatch>`), siga las mismas reglas que en el caso de una versión regular. Utilice una versión de la expresión regular del contenedor únicamente si está seguro de que sus expresiones regulares están fuertemente expresadas.
- Debido a un error en las etapas tempranas del desarrollo de Apache, el control proxy se sigue llevando a cabo con el contendor `<Directory>`, a pesar de que el contenedor `<Location>` es más apropiado. Posiblemente esto se corregirá en la próxima versión. Sin embargo, no se trata de un grave problema, únicamente hace que las cosas sean más difíciles de conceptualizar.

El contenedor `<VirtualHost>` se discute más adelante en otra sección de este libro.

### **<Directory>**

Las etiquetas de los contenedores `<Directory>` y `</Directory>` se utilizan para encerrar un grupo de directivas que se aplican únicamente al directorio

nombrado y a sus subdirectorios. Se puede utilizar cualquier directiva que esté permitida en un contexto de directorio.

El argumento puede ser un fully qualified pathname (nombre completo de la máquina incluido su dominio).

**Sintaxis:** <Directory directory> ... </Directory>

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

En el siguiente ejemplo se utiliza el directorio /www/mycompany/public/htdocs/download como un fully qualified pathname. Este ejemplo permite realizar un índice de directorios en este directorio.

```
<Directory /www/mycompany/public/htdocs/download>
    Options +Indexes
</Directory>
```

También puede utilizar caracteres comodín en la determinación de la ruta. En el siguiente ejemplo, la interrogación ? corresponderá a un solo carácter:

```
<Directory /www/mycompany/public/htdocs/downloa?>
    Options +Indexes
</Directory>
```

Por lo tanto, los directorios como /www/mycompany/public/htdocs/download y /www/mycompany/public/htdocs/downloaD serán emparejados. También puede utilizar el \* (asterisco) para emparejar cualquier secuencia de caracteres distintos del carácter / (barra). Se puede añadir el carácter ~ para expresiones regulares extendidas. Por ejemplo:

```
<Directory ~ "^\w{www/.*/"}>
```

corresponderá a cualquier subdirectorio bajo /www/. Tenga en cuenta que es posible que esos contenedores <Directory> basados en expresiones, no se pueden aplicar hasta que no se apliquen todos los contenedores <Directory> y los archivos .htaccess normales (es decir, sin expresiones regulares). Entonces, se probarán todas las expresiones regulares en el orden en el que aparecen en el archivo de configuración.

**TRUCO:** Ver el apéndice B para obtener una explicación detallada sobre las expresiones regulares.

Si especifica más de un contenedor <Directory> para el mismo espacio de directorio, se aplicará primero el contenedor <Directory> de menor alcance. Por ejemplo:

```
<Directory /www>
    AllowOverride None
</Directory>
<Directory ~ "/www/mycompany/public/htdocs/*">
    AllowOverride FileInfo
</Directory>
```

De acuerdo con lo dicho, cuando llegue una solicitud a /www/mycompany/public/htdocs/somefile.cvs, Apache invalidará el archivo de control de acceso a nivel de directorios (.htaccess) para /www y lo capacitará para /www/mycompany/public/htdocs. Además, aceptará cualquier directiva FileInfo como DefaultType desde el interior del archivo /www/mycompany/public/htdocs/.htaccess.

## **<DirectoryMatch>**

El contenedor DirectoryMatch es prácticamente igual que el contenedor <Directory> excepto en que toma una expresión regular como argumento y no necesita el carácter ~. <DirectoryMatch> y </DirectoryMatch> se utilizan para encerrar un grupo de directivas que se aplican únicamente al directorio nombrado y a sus subdirectorios.

**Sintaxis:** <DirectoryMatch regex> ... </DirectoryMatch>

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

En el siguiente ejemplo, encontrará correspondencias con todos los subdirectorios de /www/mycompany/public/htdocs que tengan exactamente ocho letras mayúsculas en el nombre; por lo tanto, /www/mycompany/public/htdocs/AAAABBBA/ se corresponderá con la expresión regular anterior.

```
<DirectoryMatch "^[^/www/mycompany/public/htdocs/[A-Z]{8}/*">
```

Ver el apéndice B para obtener más detalles sobre las expresiones regulares.

## **<Files>**

Para controlar el acceso mediante el nombre de archivo, tiene que utilizar el contenedor Files. Las secciones <Files> se procesan en el orden en el que aparecen en el archivo de configuración, después de que se lean las secciones <Directory> y los archivos .htaccess, pero antes de que se lean las secciones <Location>.

**Sintaxis:** <Files nombre de archivo> ... </Files>

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual, nivel de directorios

El argumento *nombre de archivo* debe incluir el nombre de archivo, o una cadena comodín, en la que ? corresponda a cualquier carácter, y \* corresponda a cualquier secuencia de caracteres excepto el carácter /. Utilizando el carácter, puede ser capaz de revisar expresiones regulares extendidas en el argumento. Por ejemplo:

```
<Files ~ "\.(zip|tar|tgz|arj|zoo)$">
```

corresponderá a cualquier archivo con la extensión .zip, .tar, .tgz, .arj o .zoo a diferencia de las secciones <Directory> y <Location>, las secciones <Files> se pueden utilizar dentro de los archivos .htaccess. Cuando utilizamos estas secciones dentro de un archivo .htaccess, no necesita adjuntar el nombre de la ruta, porque un archivo .htaccess sólo se aplica al directorio en el se ha encontrado.

## <FilesMatch>

El contenedor FilesMatch es exactamente el mismo que el contenedor <Files>, excepto en que toma una expresión regular como argumento.

**Sintaxis:** <FilesMatch regex> ... </Files>

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual, nivel de directorios

Por ejemplo, el siguiente ejemplo coincidirá con cualquier archivo con la extensión .zip, .tar, .tgz, .arj y .zoo. Tenga en cuenta que no necesita el carácter ~ en este contenedor para utilizar una expresión regular:

```
<FilesMatch "\.( zip|tar|tgz|arj|zoo)$">
```

## <Location>

El contenedor <Location> proporciona control de acceso mediante URL. Los contenedores <Location> se procesan en el orden en el que aparecen en el archivo de configuración, después de leer los contenedores <Directory> y los archivos .htaccess.

**Sintaxis:** <Location URL> ... </Location>

**Predefinido:** ninguno

**Contexto:** servidor, host virtual

El argumento URL no necesita el http://servername. Puede utilizar caracteres comodín como una ? (que empareja cualquier carácter) o \* (que empareja cualquier secuencia de caracteres excepto para el carácter /). También puede utilizar una expresión extendida utilizando el carácter ~ antes de la expresión. Por ejemplo, <Location ~ "/(my|your)/file"> corresponderá a URLs como /my/file o your/file.

## <LocationMatch>

El contenedor LocationMatch es idéntico al contenedor <Location>, excepto en que su argumento (URL) es una expresión regular y en que no necesita un ~ antes de la expresión. Por ejemplo, <LocationMatch "/(my|your)/file"> emparejará URLs como /my/file o your/file.

**Sintaxis:** <LocationMatch regex> ... </LocationMatch>

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

## Directivas específicas de host virtuales

Estas directivas se utilizan para crear host virtuales. Por defecto, Apache únicamente sirve al host del sitio Web especificado en la directiva ServerName. Es posible, sin embargo, hacer que Apache sirva a otros sitios Web utilizando una directiva de un contenedor de host virtuales. Tenga en cuenta que muchas de las directivas que hemos visto en la sección Directivas de configuración general también se pueden aplicar a host virtuales.

### NameVirtualHost

Si está pensando en utilizar host virtuales basados en nombres, necesita utilizar la directiva NameVirtualHost. Aunque addr puede ser el nombre del host, le recomiendo que utilice siempre una dirección IP.

**Sintaxis:** NameVirtualHost addr[:port]

**Predefinido:** ninguno

**Contexto:** configuración del servidor

Por ejemplo, para un host virtual llamado www.mycompany.com que utiliza la dirección IP 192.168.1.200, la directiva y la definición del host virtual serán:

```
NameVirtualHost 192.168.1.200  
  
<VirtualHost 192.168.1.200>  
    ServerName www.mycompany.com  
    # Aquí van otras directivas  
</VirtualHost>
```

Si tiene varios host basados en nombre en varias direcciones, ha de repetir esta directiva para cada dirección. En el listado 4.1, la primera directiva NameVirtualHost se utiliza para los host virtuales www.mycompany.com y www.friendscomany.com. El segundo contenedor se utiliza para los host virtuales www.myclient.com y www.herclient.com.

#### Listado 4.1. Directiva NameVirtualHost

```
NameVirtualHost 192.168.1.200

#
# Primer host virtual que corresponde a la directiva anterior
#
<VirtualHost 192.168.1.200>
    ServerName www.mycompany.com
    # Aquí van otras directivas
</VirtualHost>

# Segundo host virtual que corresponde a la directiva anterior
#
<VirtualHost 192.168.1.200>
    ServerName www.friendscompany.com
    # Aquí van otras directivas
</VirtualHost>

# Otra directiva NameVirtualHost para un nuevo conjunto de
# host virtuales basados en nombre que
# utilizan una IP distinta.

NameVirtualHost 192.168.1.100>

#
# Primer host virtual que corresponde a 192.168.1.100
#
<VirtualHost 192.168.1.100>
    ServerName www.myclient.com
    # Aquí van otras directivas
</VirtualHost>

#
# Segundo host virtual que corresponde a 192.168.1.100
#
<VirtualHost 192.168.1.100>
    ServerName www.herclient.com
    # Aquí van otras directivas
</VirtualHost>
```

De forma opcional, puede especificar un número de puerto en el que se podrían utilizar los host virtuales basados en nombre. Por ejemplo:

```
NameVirtualHost 192.168.1.100:8080
```

#### ServerAlias

Esta directiva le permite definir un alias para su nombre de host en su servidor principal. Cuando tiene un host virtual basado en nombre con varios nombres IP

(registros CNAME en la base de datos DNS), puede utilizar una sola definición de host virtual para servirlos a todos.

**Sintaxis:** ServerAlias host1 [host2 ...]

**Predefinido:** Ninguno

**Contexto:** Host virtual

En el siguiente ejemplo, www.sac-state.edu y www.csu.sacramento.edu son alias del host virtual www.csus.edu.

```
NameVirtualHost 192.168.1.100

<VirtualHost 192.168.1.100>
    ServerName www.csus.edu
    ServerAlias www.sac-state.edu    www.hornet.edu
</VirtualHost>
```

**TRUCO:** También puede utilizar comillas dobles o simples en la definición de los alias.

## ServerPath

La directiva ServerPath asigna los nombres de rutas de URL heredadas de un host, para utilizarlo con los host virtuales basados en nombre. Normalmente, se utiliza para soportar navegadores que no son compatibles con HTTP 1.1.

**Sintaxis:** ServerPath nombre ruta

**Predefinido:** ninguno

**Contexto:** host virtual

## <VirtualHost>

La directiva del contenedor `<VirtualHost>` especifica una configuración para el host virtual. Todas las directivas encontradas en `<VirtualHost>` y en `</VirtualHost>` se aplican únicamente a dicho host virtual. Se puede utilizar cualquier directiva que esté permitida en el contexto de un host virtual. Cuando un servidor recibe la solicitud de un documento en un host virtual determinado, utiliza las directivas de configuración encerradas en `<VirtualHost>`.

**Sintaxis:** <VirtualHost addr[:port] ...> ... </VirtualHost>

**Predefinido:** ninguno

**Contexto:** configuración del servidor

Para especificar qué dirección IP o qué nombre IP se va a utilizar en un host determinado, puede utilizar:

- Una dirección IP. Por ejemplo:

```
<VirtualHost 192.168.1.100>
    # las directivas van aquí
</VirtualHost>
```

- Una dirección IP con un número de puerto. Por ejemplo:

```
<VirtualHost 192.168.1.100:8080>
    # la directiva va aquí
</VirtualHost>
```

- Varias direcciones IP. Por ejemplo:

```
<VirtualHost 192.168.1.100 192.168.1.105>
    # las directivas van aquí
</VirtualHost>
```

- \* Varias direcciones IP con números de puerto. Por ejemplo:

```
<VirtualHost 192.168.1.100:8000 192.168.1.105:10000>
    # las directivas van aquí
</VirtualHost>
```



Se puede utilizar el nombre especial `_default_` si este host virtual corresponde a cualquier dirección IP que no se encuentre explícitamente en una lista en otro host virtual. En ausencia de cualquier host virtual `_default_`, se utiliza la configuración del servidor principal, que consiste en todas las definiciones que se encuentran fuera de la sección `VirtualHost`, cuando no hay ninguna coincidencia.

Si no se especifica un puerto, entonces el número de puerto por defecto es el mismo puerto que se encuentra en la directiva `Port` del servidor principal. También puede especificar que \* empareje todos los puertos en estas direcciones.

## Directivas de registro

Las transacciones de registro en los servidores son un deber en cualquier sistema que esté ejecutando Apache. Los registros en los servidores proporcionan información de gran valor, como quién accede a su sitio(s) Web, a qué páginas acceden y qué errores genera el servidor.

## LogLevel

La directiva LogLevel asigna el mensaje de registro almacenado en el archivo de registro de errores. Cuando especifica un nivel de registro, todos los mensajes de nivel superior se escriben en un registro. Por lo que, si especifica un nivel crit, entonces únicamente están registrados los errores emerg, alert y crit.

**Sintaxis:** LogLevel nivel

**Predefinido:** LogLevel error

**Contexto:** configuración del servidor, host virtual

La tabla 4.2 muestra los niveles disponibles (en orden descendente) con sus respectivos significados.

**Tabla 4.2.** Niveles en LogDirective

Nivel	Significado
Emerg	Situación de extrema emergencia.
Alert	Se requiere una acción inmediata.
Crit	Errores críticos.
Error	Condiciones de error.
Warn	Mensajes de alerta.
Notice	Noticias de varios tipos.
Info	Mensajes de información.
Debug	Mensajes de depuración de errores.

La directiva ErrorLog especifica el nombre del archivo de registro utilizado para registrar mensajes que produce el servidor. Si el nombre de archivo no comienza con una barra (/), entonces asume que se refiere al ServerRoot.

**Sintaxis:** ErrorLog nombre archivo

**Predefinido:** ErrorLog logs/error\_log

**Contexto:** configuración del servidor, host virtual

Si necesita desactivar el registro de errores, puede utilizar lo siguiente:

```
ErrorLog /dev/null
```

**NOTA:** Es muy importante que tenga los permisos adecuados para el directorio de registros del servidor. De lo contrario, Apache

**(especificado por la directiva User) tiene permiso de acceso a la lectura y a la escritura. Permitir a alguien más escribir en este directorio puede crear agujeros potenciales de seguridad.**

## PidFile

Utilizando la directiva `PidFile`, puede pedirle a Apache que escriba el proceso ID (o PID) en el servidor principal ID (es decir, el proceso demonio) en un archivo. Si el nombre de archivo no comienza con una barra (/), entonces asume que se refiere al `ServerRoot`. La directiva `PidFile` se utiliza únicamente en solitario.

**Sintaxis:** `PidFile nombre archivo`

**Predefinido:** `PidFile logs/httpd.pid`

**Contexto:** configuración del servidor

La principal utilización de la directiva `PidFile` es procurar que el administrador de Apache encuentre el PID principal de Apache, que es necesario para enviar señales al servidor. Por ejemplo, si el archivo PID se guarda en el directorio `/usr/local/httpd/logs`, y su nombre es `httpd.pid`, un administrador puede forzar al servidor Apache para que relea su configuración enviando una señal `SIGHUP` desde el lugar donde el usuario teclea comandos del shell (como raíz) del modo siguiente:

```
kill -HUP `cat /usr/local/httpd/logs/httpd.pid`
```

Este mismo comando hace que Apache reabra el `ErrorLog` y el `TransferLog`.

**ADVERTENCIA:** Al igual que ocurre con cualquier otro archivo de registro, asegúrese de que el archivo PID no es reescribible o incluso que no lo puede leer nadie que no sea el proceso del servidor. Para mayor seguridad, debería hacer que sólo pudiese escribir y leer en el directorio de registro el usuario del servidor Apache.

## ScoreBoardFile

La directiva `ScoreBoardFile` asigna la ruta al archivo utilizado para almacenar datos de procesos internos. Si el nombre de archivo no comienza con una barra (/), entonces se asume que se refiere a `ServerRoot`.

El proceso principal del servidor utiliza este archivo para comunicarse con los procesos hijo.

**Sintaxis:** ScoreBoardFile nombre archivo

**Predefinido:** ScoreBoardFile logs/apache\_status

**Contexto:** configuración del servidor

Si quiere averiguar si su sistema necesita este archivo, simplemente tiene que ejecutar el servidor Apache y ver si se crea un archivo en una localización determinada. Si su arquitectura de sistema necesita este archivo, entonces debe asegurarse de que este archivo no se está utilizando al mismo tiempo por más de una invocación de Apache. Además, asegúrese de que no hay otro usuario que tenga acceso a la lectura o escritura en este archivo, o incluso al directorio en el que está guardado.

**NOTA:** Como el proceso tiene que llevar a cabo salidas y entradas en el disco para comunicarse, esto podría causar potencialmente un cuello de botella en el rendimiento; por tanto, debería crear, si es posible, un disco RAM para este archivo. Consulte los manuales de su sistema operativo para obtener más detalles.

## Directivas de autentificación y de seguridad

Las directivas de autentificación y seguridad que se discuten en las siguientes secciones le permiten definir las restricciones de autentificación y acceso a su servidor Web.

Puede utilizar autentificación basada en nombre de usuario y contraseña para restringir el acceso a determinadas partes de su sitio Web. Además, puede utilizar el control de acceso basado en el nombre de usuario, en la dirección IP o en el nombre del host para asegurar que sólo tienen permitido el acceso los usuarios o los sistemas válidos a las distintas partes de su sitio Web.

### AllowOverride

La directiva AllowOverride le dice al servidor qué directivas, de las que están declaradas en un archivo .htaccess (como las especificadas por AccessFileName), pueden invalidar directivas encontradas en los archivos de configuración.

Cuando Override se fija en None, el servidor no lee el archivo especificado por FileName (.htaccess por defecto). Esto puede acelerar el tiempo de respuesta del servidor, porque el servidor no tiene que buscar un archivo FileName específico para cada solicitud (ver la sección AccessFileName para obtener los detalles).

**Sintaxis:** AllowOverride opcion1 opcion2 ...

**Predefinido:** AllowOverride All

**Contexto:** directorio

Si quiere permitir el control de acceso basado en AccessFileName, puede especificar una o más de estas opciones. Las opciones para invalidar son:

- AuthConfig: permite la utilización de las directivas de autorización (como AuthDBMGroupFile, AuthDBMUserFile, AuthGroupFile, AuthName, AuthType, AuthUserFile y Require).
- FileInfo: permite la utilización de las directivas de control del tipo de documentos (como AddEncoding, AddLanguage, AddType, DefaultType, ErrorDocument y LanguagePriority).
- Indexes: permite la utilización de directivas de control de la realización de índices de directorio (como AddDescription, AddIcon, AddIconByEncoding, AddIconByType, DefaultIcon, DirectoryIndex, FancyIndexing, HeaderName, IndexIgnore, IndexOptions y ReadmeName).
- Limit: permite la utilización de directivas de control de acceso al host (Allow, Deny y Order).
- Options: permite la utilización de directivas que controlan las características específicas de directorios (Options y XBitHack).

## AuthName

La directiva AuthName asigna el nombre del campo para un recurso (como un directorio) que necesita autentificación. El navegador Web normalmente muestra este campo en una ventana de diálogo pop-up cuando teclea un nombre de usuario y una contraseña para acceder al recurso solicitado (controlado). No hay un nombre de campo por defecto.

La función principal de esta etiqueta es informar al usuario del lado del cliente sobre los recursos a los que está intentando acceder.

**Sintaxis:** AuthName "authentication\_realm\_name"

**Predefinido:** ninguno

**Contexto:** directorio, configuración en el ámbito de directorios

**Invalidar:** AuthConfig

Por ejemplo, "AuthName Secured Game Zone" le dice a los usuarios que están solicitando entrar en el área Secured Game Zone (Zona segura de juegos) de un sitio. Tenga en cuenta que para que esta directiva funcione, debe ir

acompañada de las directivas `AuthType`, `Require`, `AuthUserFile` y `AuthGroupFile`.

## AuthType

La directiva `AuthType` selecciona el tipo de autentificación de usuario para un directorio. Actualmente, en Apache sólo están implementados los tipos de autentificación HTTP Basic o la autentificación Digest. La autentificación Basic no se debe utilizar para necesidades serias; la contraseña y el nombre de usuario se trasmitten en texto.

La contraseña y el nombre de usuario se retransmiten para cada solicitud siguiente que integre en el mismo directorio restringido o en sus mismos subdirectorios. La autentificación Digest es más segura que la Basic pero no está disponible en todos los navegadores Web. Ver el capítulo 7 para obtener más detalles.

La directiva `AuthType` debe ir acompañada de `AuthName` y necesita otras directivas como `AuthUserFile` y `AuthGroupFile`, para funcionar.

**Sintaxis:** `AuthType Basic | Digest`

**Predefinido:** ninguno

**Contexto:** directorio, configuración en el ámbito de directorios

**Invalidar:** `AuthConfig`

## HostNameLookups

La directiva `HostNameLookups` instruye a Apache para activar o desactivar una búsqueda DNS para cada solicitud. Cuando está activada, Apache almacena el nombre del host del cliente en la variable de entorno `REMOTE_HOST` de cada proceso CGI y SSI que se ejecuta.

**Sintaxis:** `HostNameLookups on | off | double`

**Predefinido:** `HostNameLookups off`

**Contexto:** servidor, host virtual, directorio, configuración en el ámbito de directorios

Los valores `on` (encendido) y `off` (apagado) significan exactamente lo que implican sus nombres. El valor `double` indica una búsqueda DNS doble inversa, es decir, una vez que se ha realizado una búsqueda inversa, tiene lugar una directa sobre el resultado.

Al menos una de las direcciones IP en la búsqueda directa debe coincidir con la dirección original. Sin embargo, los procesos CGI y SSI no obtienen los resultados de las búsquedas DNS dobles.

**NOTA:** No tiene que preocuparse el valor que asigne a esta directiva, cuando se utiliza mod\_access para controlar el acceso por nombre al host, tiene lugar una búsqueda doble inversa, que no es rápida pero es necesaria para garantizar la seguridad.

Le recomiendo que mantenga el valor por defecto en esta directiva. Esta decisión eliminará gran cantidad de tráfico DNS innecesario desde la red. Si quiere activarlo para que sus archivos de registro contengan nombres IP en lugar de direcciones IP, debería considerar otra opción, como ejecutar la utilidad logresolve para resolver la dirección IP y los nombres IP.

## IdentityCheck

La directiva IdentityCheck le dice a Apache que registre los nombres de usuario remotos interactuando con el proceso identd (identificación demonio) del usuario remoto, o con un servidor compatible con RFC1413. Pocas veces se trata de una directiva útil porque no funciona en todos los sistemas. La mayor parte de los sistemas no ejecutan procesos identd para proporcionar identificaciones remotas de usuarios a servidores remotos.

**Sintaxis:** IdentityCheck On | Off

**Predefinido:** IdentityCheck Off

**Contexto:** configuración del servidor, host virtual, directorio, configuración en el ámbito de directorio

**ADVERTENCIA:** Si decide utilizar esta directiva en su configuración, procure que la información que registre no sea revelada en ningún sentido excepto para el seguimiento. Esta directiva puede aumentar los problemas de rendimiento porque el servidor tiene que realizar una verificación por cada solicitud. Además, cuando un usuario remoto no está proporcionando un servicio identd o está tras un firewall o un proxy, el proceso de verificación tiene que agotar el tiempo de espera.

## <Limit>

La directiva del contenedor <Limit> se utiliza para encerrar un grupo de directivas de control de acceso, que se aplicarán únicamente a los métodos HTTP especificados. Los nombres de los métodos pueden ser uno o más de los siguientes: GET, POST, PUT, DELETE, CONNECT y OPTIONS. Si se utiliza GET, restringirá también las solicitudes HEAD. Si quiere limitar todos los métodos, no

incluya ningún método en la directiva <Limit>. Tenga en cuenta que este contenedor no se puede anidar, ni tampoco puede aparecer un contenedor <Directory> dentro de él. Los nombres de los métodos cambian al utilizar mayúsculas o minúsculas.

**Sintaxis:** <Limit method method ... > ... </Limit>

**Predefinido:** ninguno

**Contexto:** todos

## <LimitExcept>

La directiva del contenedor <LimitExcept> se utiliza de forma opuesta a la directiva <Limit>, <limit> limita los métodos nombrados (por ejemplo, arguments) y <LimitExcept> limita todo lo que no sean argumentos. Todos los métodos que no están en la lista de argumentos están limitados.

**Sintaxis:** <LimitExcept metodo metodo ... > ... </LimitExcept>

**Predefinido:** ninguno

**Contexto:** todos

En el siguiente ejemplo, se aplica el límite a todos los métodos HTTP excepto a GET.

```
<LimitExcept GET>
    # directivas
</LimitExcept>
```

## LimitRequestBody

La directiva LimitRequestBody le permite asignar un límite en el tamaño de la solicitud HTTP que Apache servirá. El límite por defecto es 0, que significa ilimitado. Puede asignar este límite de 0 a 2147483647 (2GB).

**Sintaxis:** LimitRequestBody bytes

**Predefinido:** LimitRequestBody 0

**Contexto:** configuración del servidor, host virtual, directorio, configuración en el ámbito de directorio

Se recomienda asignar un límite, sólo si tiene experiencia en evitar ataques basados en la denegación de servicios HTTP, en los que intentan sobrecargar el servidor con grandes solicitudes HTTP. Se trata de una directiva útil a la hora de mejorar la seguridad en el servidor.

## **LimitRequestFields**

La directiva `LimitRequestFields` le permite limitar el número de campos de cabeceras de solicitudes permitidos en una sola solicitud HTTP. Este límite se puede encontrar entre 0 y 32767 (32K). Esta directiva le puede ayudar a implementar las medidas de seguridad contra ataques basados en la denegación de servicios con grandes solicitudes.

**Sintaxis:** `LimitRequestFields` *número*

**Predefinido:** `LimitRequestFields` 100

**Contexto:** configuración del servidor

## **LimitRequestFieldsize**

La directiva `LimitRequestFieldsize` le permite limitar el tamaño (en bytes) de un campo de cabecera de solicitud. El tamaño por defecto es 8190 (8K) que es más que suficiente para la mayor parte de las situaciones. Sin embargo, si tiene lugar un ataque basado en la denegación de servicios HTTP, puede cambiarlo a un número más pequeño para denegar las solicitudes que excedan el límite. El valor 0 asigna el límite en ilimitado.

**Sintaxis:** `LimitRequestFieldsize` *bytes*

**Predefinido:** `LimitRequestFieldsize` 8190

**Contexto:** configuración del servidor

## **LimitRequestLine**

La directiva `LimitRequestLine` asigna el límite de tamaño de la línea de solicitudes. En realidad limita el tamaño de la URL que se puede enviar al servidor. El límite por defecto debería ser suficiente para la mayoría de las situaciones. Si tiene lugar un ataque basado en la denegación de servicios que utilice una URL muy larga para agotar los recursos en su servidor, puede reducir el límite para rechazar este tipo de solicitudes.

**Sintaxis:** `LimitRequestLine` *bytes*

**Predefinido:** `LimitRequestLine` 8190

**Contexto:** configuración del servidor

## **Require**

Apache determina qué usuarios o grupos pueden acceder a un directorio restringido utilizando la directiva `Require`. Hay tres tipos de nombres de entidad

disponibles: user, group, valid-user. Por ejemplo, require user joe jenny le dice a Apache que permita la entrada al área únicamente a joe o a jenny tras una autentificación con éxito. Sólo los usuarios nombrados pueden acceder al directorio.

**Sintaxis:** Require nombre-entidad nombre-entidad...

**Predefinido:** ninguno

**Contexto:** directorio, configuración en el ámbito de directorio

**Invalidar:** AuthConfig

A continuación tenemos un ejemplo en el que sólo los usuarios de los grupos nombrados pueden acceder al directorio, se trata de un ejemplo con un requisito de acceso basado en grupo:

```
Require group my-group your-group his-group her-group
```

Con la siguiente línea, todos los usuarios válidos pueden acceder al directorio.

```
require valid-user
```

Si la directiva require aparece en una sección <Limit>, entonces restringe el acceso a los métodos nombrados; en caso contrario, restringe el acceso para todos los métodos. Por ejemplo:

```
AuthType Basic
AuthName "Game Zone Drop Box"
AuthUserFile /www/netgames/.users
AuthGroupFile /www/ntgames/.groups

<Limit GET>
    require group coders
</Limit>
```

Si la configuración anterior se encuentra en un archivo .htaccess en un directorio, únicamente un grupo llamado coders tiene permiso de acceso al directorio para recuperar archivos mediante el método HTTP GET. Para que funcione adecuadamente la directiva Require debe ir acompañada de las directivas AuthName y AuthType, y por directivas del tipo AuthUserFile y AuthGroupFile.

Para obtener los detalles de autentificación, ver los capítulos siguientes.

## Satisfy

Si ha creado una configuración básica de autentificación HTTP en la que se utilizan las directivas Allow y Require, puede utilizar la directiva Satisfy para decirle a Apache que los requisitos de autentificación son suficientes.

**Sintaxis:** Satisfy Any | All

**Predefinido:** Satisfy all

**Contexto:** directorio, nivel de directorios

El valor de la directiva `Satisfy` puede ser `all` o `any`. Si el valor es `all`, entonces la autenticación será exitosa en el caso de que tengan éxito `Allow` y `Require`. Si el valor es `any`, entonces la autenticación será exitosa en el caso de que no lo sean ni `Allow` ni `Require`.

La directiva `Satisfy` es útil sólo si se restringe una determinada zona, tanto con contraseña y nombre de usuario como con la dirección del host cliente. En este caso, el comportamiento por defecto (`all`) requiere que el cliente pase la restricción de acceso de dirección e introduzca una contraseña y un nombre de usuario válidos. Con la opción `any`, se permite el acceso al cliente si el usuario, o bien pasa la restricción del host o bien introduce un nombre de usuario y una contraseña válidos. Esta directiva se puede utilizar para restringir el acceso a una zona utilizando contraseñas, al mismo tiempo, y de forma simultánea, le da acceso a todos los clientes de un conjunto determinado de direcciones IP (es decir, un conjunto de direcciones IP) sin pedirles que introduzcan contraseña.

## ScriptInterpreterSource

La directiva `ScriptInterpreterSource` le permite especificar el modo en que Windows encuentra el intérprete para un script. Normalmente, el intérprete de script se detecta utilizando la línea `#!` encontrada en un script. Sin embargo, asignando esta directiva para que registre, fuerza a Windows a realizar una búsqueda en el registro de las extensiones de los script para encontrar el programa solicitado (es decir, el intérprete).

**Sintaxis:** ScriptInterpreterSource Registry | Script

**Predefinido:** ScriptInterpreterSource script

**Contexto:** directorio, .htaccess

## Directivas específicas de MPM threaded

Es igual que en los MPM prefork, pero en lugar de que cada proceso hijo tenga un solo hilo, cada proceso hijo puede tener un número determinado de hilos. Como los hilos son más eficaces en cuanto a recursos que los procesos, este MPM es muy escalable. Cada hilo dentro de un proceso hijo puede servir una solicitud distinta.

Los procesos se añaden o se eliminan controlando su conteo de hilos producidos. Por ejemplo, si un proceso tiene menos hilos producidos que el valor mínimo,

se añade un nuevo proceso. De igual modo, cuando un proceso tiene un número máximo de hilos parados, es asesinado.

**NOTA:** Todos los procesos se ejecutan bajo el mismo ID de usuario y de grupo, asignado por el servidor Apache.

## CoreDumpDirectory

La directiva `CoreDumpDirectory` asigna el directorio que Apache intenta cambiar antes de que se estropee el archivo general. La localización por defecto es el directorio especificado por la directiva `ServerRoot`.

**Sintaxis:** `CoreDumpDirectory ruta-directorio`

**Predefinido:** directorio de la ruta del servidor

**Contexto:** configuración del servidor

## Group

La directiva `Group` debería utilizarse en unión con la directiva `User`. `Group` determina el grupo bajo el cual el servidor responde a las solicitudes. Para utilizar esta directiva, el servidor debe ejecutarse inicialmente como raíz. La directiva `Group` puede tener un número de grupo como valor asignado. `Group` busca nombres de grupo y sus correspondientes valores numéricos en su archivo `/etc/group`.

**Sintaxis:** `Group Unix-group`

**Predefinido:** `Group #-1`

**Contexto:** Configuración del servidor, host virtual

**NOTA:** Todas las advertencias y errores generados por la directiva Group se pueden aplicar a cada grupo definido. La directiva User más tarde se aplica a cada grupo.

## Listen

Por defecto, Apache responde a las solicitudes en todas las direcciones IP adjuntas al servidor, pero sólo por la dirección del puerto especificado por la directiva `Port`. La directiva `Listen` se puede utilizar para hacer esta situación más configurable. Puede utilizar la directiva `Listen` para pedirle a Apache que

responda a cierta dirección IP, a una combinación de dirección IP y de puerto o, simplemente a un puerto.

**Sintaxis:** Listen [IP address:] numero-puerto

**Predefinido:** ninguno

**Contexto:** configuración del servidor

A pesar de que se puede utilizar Listen en lugar de BindAddress y Port, podría tener que utilizar la directiva Port si su servidor Apache genera URLs que se dirigen a él mismo.

Se pueden utilizar varias directivas Listen para especificar el número de direcciones y puertos a los que escuchar. El servidor responderá a las solicitudes desde cualquiera de las direcciones y puertos de la lista. Por ejemplo, para conseguir que el servidor acepte conexiones del puerto 80 y del puerto 8080, utilice:

```
Listen 80  
Listen 8080
```

El siguiente ejemplo, consigue que Apache acepte conexiones en dos direcciones IP y en dos números de puerto:

```
Listen 192.168.1.100:80  
Listen 192.168.1.101:8080
```

## ListenBacklog

La directiva ListenBacklog le permite tomar medidas de defensa contra un ataque a la seguridad llamado Denegación de Servicios (DOS), permitiéndole que asigne la longitud máxima de la cola de conexiones pendientes. Auméntelo en caso de que detecte que se encuentra ante un ataque TCP SYN flood (DOS); en caso contrario, déjelo como está.

**Sintaxis:** ListenBacklog pendientes

**Predefinido:** ListenBacklog 511

**Contexto:** configuración del servidor

## LockFile

Si compila Apache con la opción USE\_FCNTL\_SERIALIZED\_ACCEPT o la opción USE\_FLOCK\_SERIALIZED\_ACCEPT, se utiliza un archivo de bloqueo. Puede utilizar la directiva LockFile para asignar la ruta al nombre de archivo del archivo de bloqueo.

Asegúrese de que únicamente el servidor Apache tiene acceso para lectura y escritura en el archivo.

**Sintaxis:** LockFile nombre archivo

**Predefinido:** LockFile logs/accept.lock

**Contexto:** configuración del servidor

**NOTA:** Almacenar el archivo de bloqueo en una partición Network File System (NFS) puede causar problemas con la sincronización de los servidores.

Siempre se recomienda almacenar el archivo de bloqueo en la misma partición que el archivo httpd.conf.

Y también es recomendable no almacenar el archivo de bloqueo en un directorio que esté siendo monitoreado por el sistema de administración de archivos.

## MaxClients

La directiva MaxClients limita el número de solicitudes simultaneas que Apache puede servir. Como Apache utiliza un servidor hijo para cada solicitud, éste es también el límite efectivo para el número de servidores hijo que pueden existir al mismo tiempo.

**Sintaxis:** MaxClients numero

**Predefinido:** MaxClients 256

**Contexto:** configuración del servidor

El límite por defecto es realmente el límite asignado en el archivo `httpd.h` en la distribución fuente de Apache. Esta asignación debería servir para los sitios con carga de normal a moderada. Los programadores de Apache ponen este límite de hardware ahí por dos razones: no quieren que el servidor quiebre el sistema rellenando alguna tabla kernel, y este límite máximo mantiene el archivo marcador lo bastante pequeño como para que se pueda leer con facilidad. Cuando el servidor alcanza el máximo conteo de solicitudes, deja las solicitudes entrantes en estado de espera hasta que queda libre para darles servicio.

## MaxRequestsPerChild

Apache lanza un proceso hijo para servir una solicitud; sin embargo, un servidor hijo puede procesar varias solicitudes. El número de solicitudes que puede procesar un servidor hijo está limitado por la directiva MaxRequestsPerChild.

**Sintaxis:** MaxRequestsPerChild numero

**Predefinido:** MaxRequestsPerChild 0

**Contexto:** configuración del servidor

Tras servir el máximo número de solicitudes, el proceso hijo termina. Si MaxRequestsPerChild es 0, entonces el proceso nunca termina. Si sospecha que hay bibliotecas en su sistema operativo (por ejemplo, Solaris) que tienen código de filtrado de memoria, debería asignarle a esta directiva un valor distinto de cero. Esto le permite definir un ciclo vital para un proceso hijo, reduciendo las posibilidades de que un proceso consuma memoria filtrada y de que poco a poco gaste la memoria disponible. Además le proporciona una pequeña carga media para su sistema, porque la carga relacionada con Apache se reduce a medida que su servidor Web está menos ocupado.

## MaxSpareThreads

La directiva MaxSpareThreads fija el número máximo de hilos parados. El MPM threaded trata con los hilos parados en la base de un servidor, que significa que si hay demasiados hilos parados en el servidor, comienza a destruir procesos hijo hasta que el número de hilos parados es menor que el número que se especifica aquí.

**Sintaxis:** MaxSpareThreads número

**Predefinido:** MaxSpareThreads 10 (para MPM Perchild) o 500 (para MPM threaded)

**Contexto:** configuración del servidor

El MPM perchild cuenta los hilos parados en la base de cada hijo, lo que significa que si hay demasiados hilos parados en un hijo, se destruye el hijo hasta que la cuenta de hilos por hijo sea menor que el número especificado con la directiva MaxSpareThreads.

## MinSpareThreads

La directiva MinSpareThreads determina el mínimo número de hilos parados. El MPM threaded negocia con los hilos parados en la base de un servidor, lo que significa que cuando hay menos hilos parados que el número especificado aquí, Apache crea nuevos procesos hijo para alcanzar este número.

**Sintaxis:** MinSpareServers número

**Predefinido:** MaxSpareThreads 5 (para MPM Perchild) o 250 (para MPM threaded)

**Contexto:** configuración del servidor

El MPM perchild maneja el conteo de hilos parados en la base de cada hijo; por lo tanto, cuando un hijo tiene un número menor de hilos que los que se especifican aquí, el servidor crea hilos nuevos dentro de esos procesos hijo.

## SendBufferSize

La directiva `SendBufferSize` fija el tamaño de buffer TCP enviado en el número de bytes especificado. En una red de gran rendimiento, asignar a esta directiva un valor más alto que el valor por defecto del sistema operativo aumentará el rendimiento del servidor.

**Sintaxis:** `SendBufferSize bytes`

**Predefinido:** ninguno

**Contexto:** configuración del servidor

## StartServers

La directiva `StartServers` fija el número de procesos hijo del servidor Apache que se crean en la puesta en marcha. El número de procesos hijo Apache necesarios para un determinado periodo de tiempo se controla de forma dinámica. El servidor principal de Apache (el proceso demonio) lanza un proceso hijo nuevo a medida que encuentra mayor carga de solicitudes.

Las directivas `MinSpareServers`, `MaxSpareServers` y `MaxClients` controlan el número real de procesos hijo. Por lo tanto, tiene poco que ganar ajustando este parámetro.

**Sintaxis:** `StartServers número`

**Predefinido:** `StartServers 5`

**Contexto:** configuración del servidor

**NOTA:** La directiva `StartServers` es útil únicamente cuando el servidor Apache se ejecuta como un servidor autónomo. En otras palabras, necesita tener fijado `ServeType` autónomo para que esta directiva sea efectiva.

**NOTA:** Cuando ejecutamos Microsoft Windows, esta directiva determina el número total de procesos hijo en ejecución. Como la versión Windows de Apache es multihilo, un proceso maneja todas las solicitudes. El resto de los procesos están en reserva hasta que el proceso principal muera.

## ThreadsPerChild

La versión Windows de Apache es un servidor multihilo. La directiva `ThreadsPerChild` le dice al servidor cuántos hilos debería utilizar. También determina el número máximo de conexiones que puede manejar el servidor en un momento dado. Por lo tanto, este valor debería asignarse en un valor razonablemente elevado para permitir el número máximo de éxitos.

**Sintaxis:** `ThreadsPerChild numero`

**Predefinido:** `ThreadsPerChild 50`

**Contexto:** configuración del servidor (Windows)

## User

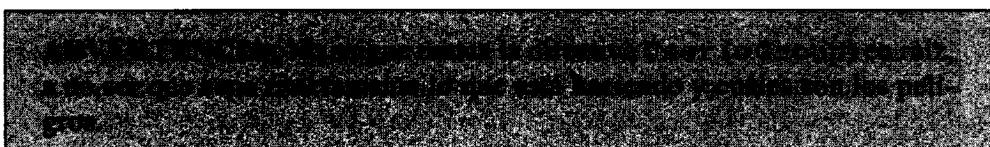
La directiva `User` asigna el ID de usuario que utilizan los hijos Apache que sirven solicitudes HTTP. Una vez que se inicia el servidor Apache, lanza procesos hijo para responder a las solicitudes. Sin embargo, estos procesos hijo se ejecutan como raíz. El proceso padre de Apache (a menudo llamado demonio) cambia el ID del usuario del proceso hijo a cualquiera de los asignados en la directiva `User`, siempre que se trate de un ID de usuario válido.

**Sintaxis:** `User unix-userid`

**Predefinido:** `User #-1`

**Contexto:** configuración del servidor, host virtual

Si no inicia el servidor como usuario raíz, no podrá cambiar el ID de usuario especificado por la directiva `User`, y en lugar de eso, continuará ejecutándose como el usuario original. Si comienza el servidor como raíz, entonces es normal que el proceso padre de Apache se quede ejecutándose como raíz; sin embargo, ejecuta el proceso hijo como el usuario especificado por la directiva `User`.



También puede utilizar los números ID de los usuarios, que puede encontrar normalmente en el archivo `/etc/password`. Si piensa utilizar un valor numérico en lugar del nombre de usuario real, el número debe ir precedido del signo `#`. Muchos administradores de Apache utilizan el usuario por defecto `nobody` para sus sitios Web. Este usuario no está disponible en todos los sistemas Unix, y además no siempre es deseable. Recomiendo encarecidamente que emplee un solo

ID de usuario y de grupo (ver la directiva `Group`) en su servidor Apache. El hacerlo le dará un mayor control sobre los accesos al servidor. El ID que decida utilizar para los procesos hijo de Apache debería tener muy pocos privilegios. No debería permitirse el acceso a archivos que no tiene intención de mostrar fuera de su sitio Web, y de igual modo, el usuario no debería ser capaz de ejecutar aplicaciones que no tienen sentido para las solicitudes HTTP.

**NOTA:** El uso de esta directiva en el contexto de `<VirtualHost>` requiere una configuración apropiada del software CGI. La directiva `CGIPath` (`<VirtualHost>`, únicamente afecta al directorio que contiene el script CGI). Las solicitudes relacionadas con CGI se ejecutan procesando con el usuario especificado en la directiva `User`. Además, la directiva `User principal` no se puede usar directamente en el contexto de `<VirtualHost>`.

## Directiva específicas de MPM `perchild`

En este modelo MPM, se inicia un conjunto de procesos hijo con un número determinado de hilos. A medida que aumenta la carga de solicitudes, los procesos añaden nuevos hilos según los van necesitando. Cuando se reduce el conteo de solicitudes, los procesos disminuyen sus conteos de hilos utilizando un máximo y un mínimo asignado de conteo.

### `AssignUserID`

La directiva `AssignUserID` asigna un nombre de usuario y un nombre de grupo a un host virtual.

**Sintaxis:** `AssignUserID nombreusuario nombregrupo`

**Predefinido:** ninguno

**Contexto:** host virtual

Por ejemplo, a continuación podemos ver cómo un host virtual llamado `www.afactcat.com` es asignado a un usuario `mrbert` y a un grupo llamado `wheel`. Debe utilizar `ChildPerUserID` para especificar el número de procesos hijo que puede servir este host virtual.

```
<VirtualHost 192.168.1.100>
  ServerName www.afactcat.com
  AssignUserID mrbert wheel
#
#
```

```
# Aquí van otras directivas  
#  
</VirtualHost>
```

**NOTA:** `username` y `groupname` deben existir en su sistema. Por ejemplo, en un sistema Linux `username` y `groupname` deben estar en los archivos `/etc/passwd` y `/etc/group`, respectivamente.

## ChildPerUserID

La directiva `ChildPerUserID` asigna un número de procesos hijo para un nombre de usuario y un nombre de grupo dados para un host virtual.

**Sintaxis:** `ChildPerUserID` `númerodehijos` `nombreusuario` `nombregrupo`

**Predefinido:** ninguno

**Contexto:** host virtual

En el siguiente ejemplo, el host virtual `www.afatcat.com` será servido por los procesos hijo de Apache que se ejecutan bajo el nombre de usuario `mrbert` y el nombre de grupo `wheel`:

```
ChildPerUserID 10 mrbert wheel  
<VirtualHost 192.168.1.100>  
    ServerName www.afatcat.com  
    AssignUserID mrbert wheel  
    #  
    # Aquí van otras directivas  
    #  
</VirtualHost>
```

**NOTA:** `username` y `groupname` deben existir en su sistema para que Apache asigne el nombre de usuario. Por ejemplo, en un sistema Linux `username` y `groupname` deben existir en los archivos `/etc/passwd` y `/etc/group`, respectivamente.

## ConnectionStatus

La directiva `ConnectionStatus` determina si la información del estado se almacena internamente o no. Cuando se le asigna el valor `on`, los módulos que utilizan la información del estado funcionarán adecuadamente.

**Sintaxis:** ConnectionStatus On | Off

**Predefinido:** ConnectionStatus On

**Contexto:** configuración del servidor

## CoreDumpDirectory

Ver la directiva `CoreDumpDirectory` en la sección "Directivas específicas de MPM threaded".

## Group

Ver la directiva `Group` en la sección "Directivas específicas de MPM threaded".

## Listen

Ver la directiva `Listen` en la sección "Directivas específicas de MPM threaded".

## ListenBacklog

Ver la directiva `ListenBacklog` en la sección "Directivas específicas de MPM threaded".

## LockFile

Ver la directiva `LockFile` en la sección "Directivas específicas de MPM threaded".

## MaxRequestsPerChild

Ver la directiva `MaxRequestsPerChild` en la sección "Directivas específicas de MPM threaded".

## MaxSpareThreads

Ver la directiva `MaxSpareThreads` en la sección "Directivas específicas de MPM threaded".

## MaxThreadsPerChild

La directiva `MaxThreadsPerChild` determina el número máximo de hilos por hijo. El número por defecto es el límite de hardware. Si desea cambiarlo tiene

que modificar el archivo de cabecera apropiado. Ver la tabla 2.3 del capítulo 2 para obtener los detalles.

**Sintaxis:** MaxThreadsPerChild number

**Predefinido:** MaxThreadsPerChild 64

**Contexto:** configuración del servidor

## MinSpareThreads

Ver la directiva MinSpareThreads en la sección "Directivas específicas de MPM threaded".

## NumServers

La directiva NumServers determina el número de procesos hijo simultáneos que crea Apache.

**Sintaxis:** NumServers number

**Predefinido:** NumServers 2

**Contexto:** configuración del servidor

El MPM per-child utiliza el valor asignado por esta directiva para determinar el número de hijos simultáneos. El valor por defecto 2 no es el apropiado en todos los sistemas.

Puede cambiarlo a un valor más alto como 15 o 20. Cuanto mayor es el número, más procesos Apache se producen, lo que significa que se pueden manejar más conexiones simultáneas.

## PidFile

Ver la directiva PidFile en la sección "Directivas específicas de MPM threaded".

## ScoreBoardFile

Ver la directiva ScoreBoardFile en la sección "Directivas específicas de MPM threaded".

## SendBufferSize

Ver la directiva SendBufferSize en la sección "Directivas específicas de MPM threaded".

## **StartThreads**

La directiva `StartThreads` determina el conteo inicial de hilos por hijo. Como los conteos de hijos se controlan dinámicamente, asignar un número superior que el asignado por defecto no suele ser necesario.

**Sintaxis:** `StartThreads` `número`

**Predefinido:** `StartThreads` `5`

**Contexto:** configuración del servidor

## **User**

Ver la directiva `User` en la sección "Directivas específicas de MPM threaded".

## **Directivas específicas de MPM**

Este es el MPM para todas las versiones de la plataforma Windows, incluido Windows NT/2000/XP y Windows 9x/ME. Este módulo es multihilo; utilizando este módulo Apache creará un proceso padre y un proceso hijo. El proceso hijo crea todos los hilos que sirven la solicitud. Este módulo saca partido de algunas llamadas a funciones nativas sólo de Windows, lo que le permite funcionar mejor que versiones anteriores del servidor Apache en la plataforma Windows.

## **CoreDumpDirectory**

Ver la directiva `CoreDumpDirectory` en la sección "Directivas específicas de MPM threaded".

## **Listen**

Ver la directiva `Listen` en la sección "Directivas específicas de MPM threaded".

## **ListenBacklog**

Ver la directiva `ListenBacklog` en la sección "Directivas específicas de MPM threaded".

## **MaxRequestsPerChild**

Ver la directiva `MaxRequestsPerChild` en la sección "Directivas específicas de MPM threaded".

## **PidFile**

Ver la directiva `PidFile` en la sección "Directivas específicas de MPM threaded".

## **SendBufferSize**

Ver la directiva `SendBufferSize` en la sección "Directivas específicas de MPM threaded".

## **ThreadsPerChild**

Ver la directiva `ThreadsPerChild` en la sección "Directivas específicas de MPM threaded".

# **Directivas específicas de MPM prefork**

El MPM prefork crea un grupo de procesos hijo para servir solicitudes. Cada proceso hijo tiene un solo hilo. Por ejemplo, si Apache inicia 30 procesos hijo, puede servir 30 solicitudes simultáneamente. Si algo sale mal y el proceso hijo muere, únicamente se pierde una solicitud. El número de procesos hijo se controla utilizando un mínimo y un máximo fijos. Cuando el número de solicitudes aumenta, se añade un nuevo proceso hijo hasta que se alcanza el máximo. De igual modo, cuando falla la solicitud, se eliminan todos los procesos hijo extra.

## **CoreDumpDirectory**

Ver la directiva `CoreDumpDirectory` en la sección "Directivas específicas de MPM threaded".

## **Group**

Ver la directiva `Group` en la sección "Directivas específicas de MPM threaded".

## **Listen**

Ver la directiva `Listen` en la sección "Directivas específicas de MPM threaded".

## **ListenBacklog**

Ver la directiva `ListenBacklog` en la sección "Directivas específicas de MPM threaded".

## **LockFile**

Ver la directiva `LockFile` en la sección "Directivas específicas de MPM threaded".

## **MaxClients**

Ver la directiva `MaxClients` en la sección "Directivas específicas de MPM threaded".

## **MaxRequestsPerChild**

Ver la directiva `MaxRequestsPerChild` en la sección "Directivas específicas de MPM threaded".

## **MaxSpareServers**

Esta directiva le permite determinar el número de procesos hijo de Apache inactivos en su servidor.

**Sintaxis:** `MaxSpareServers` número

**Predefinido:** `MaxSpareServers` 10

**Contexto:** configuración del servidor

Si el número de procesos hijo de Apache inactivos excede el número máximo especificado por la directiva `MaxSpareServers`, entonces el proceso padre asesina el exceso de procesos. Sólo es necesario ajustar este parámetro en el caso de sitios realmente visitados. A no ser que sepa lo que hace, no cambie el valor por defecto.

## **MinSpareServers**

La directiva `MinSpareServers` determina el número que deseamos tener de procesos hijo inactivos en el servidor. Un proceso inactivo es aquel que no está manejando ninguna solicitud. Si hay menos procesos Apache inactivos que el número especificado en la directiva `MinSpareServers`, entonces el proceso padre crea un nuevo hijo a una velocidad máxima de 1 por segundo. Sólo es necesario ajustar este parámetro en el caso de sitios realmente visitados. A no ser que sepa lo que hace, no cambie el valor por defecto.

**Sintaxis:** `MinSpareServers` número

**Predefinido:** `MinSpareServers` 5

**Contexto:** configuración del servidor

## **PidFile**

Ver la directiva `PidFile` en la sección "Directivas específicas de MPM threaded".

## **ScoreBoardFile**

Ver la directiva `ScoreBoardFile` en la sección "Directivas específicas de MPM threaded".

## **SendBufferSize**

Ver la directiva `SendBufferSize` en la sección "Directivas específicas de MPM threaded".

## **StartServers**

Ver la directiva `StartServers` en la sección "Directivas específicas de MPM threaded".

## **User**

Ver la directiva `User` en la sección "Directivas específicas de MPM threaded".





# 5 Módulos Apache

---

## En este capítulo

1. Aprendemos a utilizar los módulos de entorno.
2. Aprendemos a utilizar los módulos de control de acceso y autentificación.
3. Aprendemos a utilizar los módulos de generación de contenido dinámico.
4. Aprendemos a utilizar los módulos de listado de directorios.
5. Aprendemos a utilizar los módulos de tipo de contenido.
6. Aprendemos a utilizar los módulos de generación de contenido dinámico.
7. Aprendemos a utilizar los módulos de cabecera de respuesta.
8. Aprendemos a utilizar los módulos de información y registro de servidores.
9. Aprendemos a utilizar los módulos de integración de URL.
10. Aprendemos a utilizar otros módulos.

Anteriormente, hemos discutido las directivas del módulo general y de los módulos multiprocceso (MPM). Apache ofrece muchas más directivas, las cuales

están disponibles en los módulos distribuidos en la fuente estándar. Estos módulos aportan una gran cantidad de funcionalidad mediante la utilización de directivas. Este capítulo discute estos módulos y sus directivas.

## Un vistazo a los módulos

En lugar de realizar una lista de todos los módulos por orden alfabético, los he agrupado basándome en su funcionalidad. Estos módulos se dividen en las categorías siguientes:

- **Relacionados con el entorno:** estas directivas le permiten asignar y reajustar las variables de entorno.
- **Control de autenticación y de acceso:** estas directivas le permiten autenticar y autorizar el acceso a usuarios, para restringir partes de su sitio Web.
- **Generación de contenido dinámico:** estas directivas le permiten ejecutar programas externos como los scripts CGI o Server Side Includes para generar contenido dinámico.
- **Configuración del tipo de contenido:** estas directivas le permiten controlar los tipos MIME de los archivos.
- **Listas de directorios:** estas directivas le permiten controlar cómo se formatean estas listas de directorios.
- **Cabecera de la solicitud:** estas directivas le permiten controlar las cabeceras de la solicitud HTTP.
- **Información y registro del servidor:** estas directivas le permiten controlar la información sobre los registros y el estado del servidor.
- **Integración URL:** estas directivas le permiten integrar, reescribir y crear alias para una URL.
- **Módulos diversos:** estas directivas le permiten controlar diversos aspectos de Apache como es el servicio proxy, el módulo WEBDEV, etc.

## Módulos relacionados con el entorno

Los módulos de la tabla 5.1, le permiten manipular el entorno que se encuentra disponible para otros módulos o programas externos como los scripts CGI (Common Gateway Interface), SSI (Server-Side Include), scripts mod\_perl, scripts PHP, servlets Java, y similares.

**Tabla 5.1.** Módulos relacionados con el entorno

Módulo	Función
mod_env	Pasa variables de entorno a programas externos como los scripts CGI y SSI.
mod_setenvif	Asigna variables de entorno condicionales utilizando información del lado del cliente.
mod_unique_id	Este módulo genera un único ID por solicitud. No tiene directivas. Este módulo está compilado por defecto. Debe configurar la fuente utilizando la opción <code>--enable-unique-id</code> con el script de configuración y compilar e instalar Apache.

## mod\_env

mod\_env está compilado por defecto. Le permite pasar variables de entorno a programas externos como los scripts CGI, SSI, scripts mod\_perl, scripts PHP, y similares. mod-env tiene las siguientes directivas.

### PassEnv

La directiva PassEnv le dice al módulo que pase una o más variables de entorno desde el propio entorno del servidor a los scripts CGI y SSI.

**Sintaxis:** PassEnv *variable* [...]

**Contexto:** configuración del servidor, host virtual

Por ejemplo, la siguiente directiva pasa las variables de entorno HOSTTYPE y PATH a programas.

### SetEnv

La directiva SetEnv asigna un valor determinado a una variable de entorno, que se pasa a scripts CGI/SSI.

Únicamente puede definir un par de variable y el valor por cada directiva SetEnv.

**Sintaxis:** SetEnv *variable* *valor*

**Contexto:** Configuración del servidor, host virtual

Por ejemplo, la siguiente directiva SetEnv envía la variable CURRENT\_CITY a SACRAMENTO:

```
SetEnv CURRENT_CITY SACRAMENTO
```

## **UnsetEnv**

La directiva `UnsetEnv` elimina una o más variables de entorno de las que se han pasado a los scriptst CGI/SSI. Se puede utilizar para asegurar que ciertas variables de entorno que están disponibles para el servidor Apache, no lo estén en sus scripts CGI.

**Sintaxis:** `UnsetEnv variable [...]`

**Contexto:** configuración del servidor, host virtual

Por ejemplo, la siguiente directiva `UnsetEnv` elimina la variable `CURRENT_STATE` de la lista de variables de entorno:

```
UnsetEnv CURRENT_STATE
```

## **mod\_setenvif**

El módulo `mod_setenvif` está compilado en Apache por defecto. Le permite crear variables de entorno personalizadas utilizando información de una solicitud HTTP. Puede utilizar esta información en la reescritura de las URL o redirigir a los usuarios a páginas distintas.

## **BrowserMatch**

La directiva `BrowserMatch` asigna y borra la asignación de variables de entorno personalizadas cuando una expresión regular coincide con un patrón encontrado en la cabecera de una solicitud HTTP `User-Agent`. Los clientes Web envían la cabecera `User-Agent` como si fueran navegadores Web, robots Web y similares.

**Sintaxis:** `BrowserMatch regex variable[=value] [...]`

**Contexto:** configuración del servidor

Por ejemplo, la siguiente directiva le asigna a una variable llamada `vbscript` el valor `no` si el campo de la cabecera de la solicitud HTTP `User-Agent` contiene la palabra `Mozilla`, y le asigna el valor `1` a una variable de entorno llamada `javascript` porque no hay ningún valor especificado para esta variable:

```
BrowserMatch ^Mozilla vbscript=no javascript
```

Vamos a ver otro ejemplo:

```
BrowserMatch IE vbscript !javascript
```

En este caso, se elimina la variable `javascript` y se le asigna el valor `1` a la variable `vbscript` si se encuentra la palabra `IE` en la cabecera de la solicitud HTTP `User-Agent`. El carácter `!` elimina la variable del entorno.

**NOTA:** Una coincidencia de expresión distingue entre mayúsculas y minúsculas.

## BrowserMatchNoCase

La directiva `BrowserMatchNoCase` es la misma que la directiva `BrowserMatch`, excepto en que proporciona coincidencia sin distinguir entre mayúsculas y minúsculas en expresiones regulares.

**Sintaxis:** `BrowserMatchNoCase regex variable[=value] [...]`

**Contexto:** configuración del servidor

Por ejemplo, la siguiente directiva busca coincidencias con `MSIE`, `msie`, `Msie` y similares:

```
BrowserMatchNoCase ^MSIE vbscript=yes
```

## SetEnvIf

Al igual que las directivas `BrowserMatch` y `BrowserMatchNoCase`, la directiva `SetEnvIf` le permite asignar y borrar variables de entorno personalizadas. En realidad, `BrowserMatch` y `BrowserMatchNoCase` son dos versiones especiales de `SetEnvIf`. Estas dos directivas únicamente pueden funcionar en la expresión regular del campo de la cabecera de la solicitud HTTP `User-Agent`, mientras que `SetEnvIf` se puede utilizar para todos los campos de las cabeceras de la solicitud, al igual que otros tipos de información relacionada con la solicitud, como el nombre del host remoto (`Remote_Host`), la dirección IP remota (`Remote_Addr`), el método de la solicitud (`Request_Method`), el URI solicitado (`Request_URI`), y el referer o la dirección desde la cual un internauta llega a su página (`Referer`).

**Sintaxis:** `SetEnvIf attribute regex envvar[=value] [...]`

**Contexto:** configuración del servidor

Por ejemplo, la siguiente directiva `SetEnvIf` asigna el valor `true` a la variable `local_user` si la cabecera de la solicitud HTTP `Remote_Host` tiene asignado el valor `yourdomain.com`.

```
SetEnvIf Remote_Host "yourdomain\.\com" local_user=true
```

## SetEnvIfNoCase

La directiva `SetEnvIfNoCase` es la misma que `SetEnvIf`, excepto en que proporciona coincidencia sin distinguir entre mayúsculas y minúsculas en expresiones regulares.

**Sintaxis:** SetEnvIfNoCase attribute regex variable [=value] [...]

**Contexto:** configuración del servidor

## mod\_unique\_id

El módulo mod\_unique\_id proporciona un toque mágico para cada solicitud que tenga garantizado ser la única a lo largo de "todas" las solicitudes bajo condiciones muy específicas. El identificador es el único a lo largo de varias máquinas en un grupo de configuración de máquinas adecuado. La variable de entorno UNIQUE\_ID está asignada al identificador para cada solicitud. No hay directivas para este módulo.

# Módulos de control de acceso y autentificación

Apache tiene una serie de módulos para llevar a cabo las tareas de autenticación y autorización.

En la mayor parte de los casos, los módulos de autentificación utilizan autenticación HTTP básica, que utiliza contraseñas de texto simple. El módulo de autorización le permite controlar el acceso a un directorio de su sitio Web mediante un nombre de usuario o una dirección IP. Los módulos que se muestran en la tabla 5.2 le permiten llevar a cabo las tareas de control de la autenticación y el acceso.

**Tabla 5.2.** Módulos de control de acceso y autentificación

Módulo	Función
mod_auth	Este es el módulo estándar de autentificación, que implementa la autentificación HTTP Basic. Ver los siguientes capítulos para obtener los detalles.
mod_auth_anon	Da acceso a los usuarios anónimos para áreas autentificadas.
mod_auth_dbm	Ofrece autentificación de usuarios utilizando archivos DBM.
mod_auth_db	Ofrece autentificación de usuarios utilizando archivos Berkeley DB.
mod_auth_digest	Este módulo implementa la autentificación Digest utilizando Message Digest 5 (MD5).

Módulo	Función
mod_access	Este módulo le permite autorizar el acceso utilizando el nombre del host o la dirección IP.

## **mod\_auth\_anon**

El módulo mod\_auth\_anon permite el acceso anónimo a áreas autenticadas. Si está familiarizado con los servidores FTP anónimos, se trata de un sistema muy parecido. Todos los usuarios pueden utilizar un ID de usuario denominado "anónimo" y su dirección de correo electrónico como contraseña para entrar. La dirección de correo electrónico introducida se almacena en los archivos de registro y se puede utilizar para seguir la pista de usuarios o para crear listas de correo electrónico de futuros clientes. Tiene que permitir este módulo utilizando la opción --enable-auth-anon en el script de configuración de la distribución de la fuente, y compilando e instalando Apache.

### **Anonymous**

Utilizando la directiva Anonymous puede especificar uno o más nombres de usuario que se pueden utilizar para acceder al área. Es una buena idea mantener el nombre de usuario "anónimo" en la lista elegida, porque está fuertemente asociado con el acceso anónimo. Si el nombre de usuario que ha elegido tiene algún espacio, asegúrese de que el nombre de usuario está rodeado por comillas.

**Sintaxis:** Anonymous user user ...

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

**Invalidar:** AuthConfig

Por ejemplo, la siguiente directiva permite a los usuarios introducir Unregistered User o anonymous como los nombre de usuario que hay que introducir en el área anónima.

```
Anonymous "Unregistered User" anonymous
```

**NOTA:** La cadena del nombre de usuario no distingue entre mayúsculas y minúsculas.

### **Anonymous\_Authoritative**

Cuando se asigna el valor on, la autenticación anónima se convierte en el esquema de autenticación completo para un directorio. En otras palabras, si

tiene varios requisitos de autentificación para un directorio y además tiene asignada esta directiva, entonces se ignorarán el resto de los métodos.

**Sintaxis:** Anonymous\_Authoritative On | Off

**Predefinido:** Anonymous\_Authoritative Off

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

**Invalidar:** AuthConfig

## **Anonymous\_LogEmail**

Cuando asignamos el valor `on` a la directiva `Anonymous_LogEmail`, cualquiera que sea la entrada en el campo de la contraseña de la ventana de autentificación del navegador, se registrará en el archivo de registro de accesos de Apache.

**Sintaxis:** Anonymous\_LogEmail On | Off

**Predefinido:** Anonymous\_LogEmail On

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

**Invalidar:** AuthConfig

## **Anonymous\_MustGiveEmail**

Cuando asignamos el valor `On`, la directiva `Anonymous_MustGiveEmail` permite que el módulo rechace solicitudes de acceso que no proporcionan las contraseñas en la forma de una dirección de correo electrónico.

**Sintaxis:** Anonymous\_MustGiveEmail On | Off

**Configuración por defecto:** Anonymous\_MustGiveEmail On

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

**Invalidar:** AuthConfig

**ADVERTENCIA:** No debe confiar en las direcciones de correo electrónico que introduce la gente en este campo. Siempre asigne el valor `On`, porque ese no es la forma correcta de introducir una dirección de correo electrónico.

## **Anonymous\_NoUserID**

Si quiere que los usuarios dejen el campo del nombre de usuario de la ventana pop-up vacío, asigne el valor `On` a la directiva `Anonymous_NoUserID`; en

caso contrario, se necesitará un nombre de usuario que coincida con el valor proporcionado en la directiva `Anonymous`.

**Sintaxis:** `Anonymous_NoUserID On | Off`

**Predefinido:** `Anonymous_NoUserID Off`

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** `AuthConfig`

## **Anonymous\_VerifyEmail**

Cuando la directiva `Anonymous_VerifyEmail` está fijada con el valor `on`, necesita que la contraseña sea una dirección válida de correo electrónico. Sin embargo, la verificación de validez es limitada. El módulo únicamente comprueba el símbolo `@` y el punto `(.)` en el campo de la contraseña. Si la contraseña introducida tiene ambos símbolos, es aceptada.

**Sintaxis:** `Anonymous_VerifyEmail On | Off`

**Predefinido:** `Anonymous_VerifyEmail Off`

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** `AuthConfig`

La siguiente configuración muestra como se pueden utilizar las directivas anteriores para proporcionar acceso anónimo a un directorio.

```
Anonymous_NoUserId off
Anonymous_MustGiveEmail on
Anonymous_VerifyEmail on
Anonymous_LogEmail on
Anonymous anonymous guest "I do not know"
AuthName Use 'anonymous' & Email address for guest entry
AuthType basic
require valid-user
```

## **mod\_auth\_dbm**

La autentificación basada en texto sencillo (utilizando `mod_auth`) es ineficaz en procesos de alta velocidad y podría afectar negativamente al rendimiento de servidor Web cuando necesitan acceso autenticado una gran cantidad de usuarios (más de 2000) a secciones Web restringidas. El módulo `mod_auth_dbm` es la mejor elección en este tipo de casos. El módulo `mod_auth_dbm` utiliza archivos DBM en lugar de archivos de texto para almacenar datos. Un archivo DBM es un tipo especial de archivo de datos que permite un acceso aleatorio a los datos almacenados más rápido.

**NOTA:** En realidad si tiene gran cantidad de usuarios, considere la autenticación basada en mod\_auth\_mysql o en Apache's :AuthDBI que se discute en próximos capítulos. La autenticación basada en DBM se recomienda únicamente si no puede utilizar una base de datos.

Un archivo DBM almacena registros de datos en un par clave=valor y mantiene una tabla de índices computada para las claves en el archivo. Utilizando la tabla de índices en un archivo DBM, es posible recuperar los registros asociados con la clave de en menos tiempo que el necesario para analizar un archivo de texto importante con miles de registros. Hay muchos DBM disponibles, siendo los más comunes GDBM, NDBM, SDBM, y Berkeley DB (BSD-DB). La tabla 5.3 muestra una lista de características para estos DBM.

**Tabla 5.3.** Características de DBM

Características	NDBM	SDBM	GDBM	BSD-DB
Restricciones de licencia	Desconocido	No	Sí	No
Independiente del orden de los bytes	No	No	No	Sí
Límites de tamaño por defecto	4K	1K	Ninguno	Ninguno
Crea archivos seguros FTP	No	Sí	Sí	Sí
Velocidad	Desconocida	Baja	Media	Rápida
Tamaño de la base de datos	Desconocido	Pequeño	Grande	Medio
Tamaño de código	Desconocido	Pequeño	Grande	Grande
La fuente se encuentra junto con Perl	No	Sí	No	No

Esta tabla está basada en la información encontrada en la documentación de Perl 5. Antes de que pueda utilizar un DBM con Apache, debe asegurarse de que el DBM elegido está instalado en su sistema. Lleve esto a cabo confirmando que la biblioteca de archivos DBM se encuentra localizada en el directorio de bibliotecas por defecto de su sistema. Va a necesitar Perl con soporte DBM. Asegúrese de que tiene la última versión de Perl compilada con el soporte del DBM elegido.

**NOTA:** Puede bajar Perl de [www.perl.com](http://www.perl.com). Es muy sencillo configurar Perl para soporte DBM. Simplemente ejecute el script de configuración, y

le indicará el soporte DBM. Por ejemplo, si elige NDBM o GDBM como su DBM, y lo tiene instalado en su sistema, entonces el script de configuración Perl le preguntará si quiere compilar Perl con -lndbm, -lgdbm, y con los indicadores de bibliotecas.

Una vez que ha instalado las bibliotecas apropiadas de DBM en su sistema, necesita configurar a Apache para que soporte archivos DBM, porque la distribución estándar de Apache no permite el soporte de DBM. Configure el soporte de Apache utilizando la opción --enable-auth-dbm en el script de configuración, y compilando e instalando Apache.

**NOTA:** Si tiene problemas compilando Apache, trate de añadir el -l your dbmname a EXTRA\_LIBS en el archivo Configuration. Por ejemplo, si está utilizando GDBM, puede añadir "-lgdbm" de modo que tengamos "EXTRA\_LIBS=-lgdbm". Recuerde que tiene que revisar el script configure y que después volver a ejecutar make. En caso de problemas, puede ser mejor intentarlo con GNU GDBM porque se utiliza en muchos más sistemas y tiene más probabilidad de encontrar ayuda en los grupos de noticias USENET.

Una vez que Apache está compilado adecuadamente para archivos DBM, puede utilizar dbmmanage para crear un archivo de usuario DBM. Comience utilizando el script de Perl dbmmanage que se encuentra en su directorio de soporte de la distribución estándar de Apache (o la distribución de la fuente) para crear un archivo de usuario basado en DBM. El script de Perl dbmmanage puede crear muchos archivos DBM populares como los archivos NDBM, GDBM y Berkley DB.

Este script puede utilizarse para crear un archivo DBM nuevo, para añadir usuarios y contraseñas, para cambiar contraseñas, para borrar usuarios o para ver información sobre usuarios. Antes de utilizar el script, debe modificar la siguiente línea en el script, para que el DBM que quiere utilizar se encuentre el primero de la lista en el array ISA:

```
BEGIN { @AnyDBM_File::ISA = qw(DB_File, NDBM_File, GDBM_file) }
```

Por ejemplo, si está pensando utilizar archivos GDBM, cambie la línea a:

```
BEGIN { @AnyDBM_File::ISA = qw(GDBM_file, DB_File, NDBM_File) }
```

Para determinar qué opciones ofrece el script, ejecútelo del siguiente modo:

```
./dbmmanage
```

Esto le muestra una línea de sintaxis con todas las opciones posibles.

Para crear un nuevo archivo DBM llamado /www/secrets/myuserdbm añadiendo un usuario llamado reader, introduzca el siguiente comando:

```
./dbmmanage /www/secrets/myuserdbm adduser reader
```

El script le pedirá que introduzca (una reentrada) una contraseña para el usuario `reader`. Una vez que lo ha hecho, añadirá el nombre de usuario y encriptará la contraseña para el archivo DBM `myuserdbm`. No utilice la opción `add` para añadir un usuario, porque no encripta la contraseña. Para ver una lista de usuarios en un archivo DBM, utilice el script siguiente:

```
./dbmmanage /path/to/your/dbmfile view
```

Una vez que tiene recompilado Apache con soporte DBM, puede utilizar el módulo `mod_auth_dbm` para proporcionar autentificación HTTP Basic basada en DBM. Observe que para Berkeley DB tiene que utilizar `mod_auth_db` en lugar de `mod_auth_dbm`.

El módulo `mod_auth_dbm` proporciona las directivas `AuthDBMUserFile`, `AuthDBMGroupFile` y `AuthDBMAuthoritative`. Vamos a ver cada una de estas directivas y algunos ejemplos en los que se utiliza el módulo `mod_auth_dbm`.

## AuthDBMUserFile

La directiva `AuthDBMUserFile` asigna el fully qualified pathname (nombre completo de la ruta de una máquina incluido, su dominio) de un archivo DBM para utilizarlo como el archivo de usuario para la autentificación DBM. El archivo contiene un par clave=valor para cada registro, en el que el nombre de usuario es la clave y la contraseña encriptada `crypt()` es el valor. Observe que cada campo en el registro está separado por dos puntos, y se puede adjuntar un dato arbitrario después del nombre de usuario inicial y los campos de contraseñas.

**Sintaxis:** AuthDBMUserFile nombre archivo

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** AuthConfig



## **AuthDbmGroupFile**

La directiva `AuthDbmGroupFile` asigna el `fully qualified pathname` (nombre completo de la ruta de una máquina incluido su dominio) del grupo de archivos

que contiene la lista de grupos de usuarios. Cada registro en el archivo es un par clave=valor, en el que la clave es el nombre de usuario y el valor es una lista de nombres separados por comas a la que pertenece el usuario.

**Sintaxis:** AuthDBMGroupFile nombre archivo

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

**Invalidar:** AuthConfig

Si prefiere no utilizar un archivo de grupos separado, puede utilizar un solo archivo DBM que proporcione tanto la información de la contraseña como la del grupo.

El formato del archivo es el siguiente:

```
DBM_Record_Key{username} = encrypted_password:  
comma_separated_group_list
```

En este caso, el nombre de usuario es la clave, y la contraseña y la lista de grupos son los dos campos del valor. Puede poner más datos en el archivo DBM tras otros dos puntos, si así lo desea; será ignorado por el módulo de autenticación.

Si utiliza un solo DBM para proporcionar tanto la información del grupo como la de la contraseña, ha de colocar las directivas AuthDBMGroup y AuthDBMUserFile en el mismo archivo.

## AuthDBMAuthoritative

Cuando utilizamos varios esquemas de autenticación como mod\_dbm y el estándar mod\_auth en el mismo directorio, puede utilizar la directiva AuthDbmGroupFile para determinar si mod\_auth\_dbm es el esquema de autenticación completo.

**Sintaxis:** AuthDBMAuthoritative On | Off

**Predefinido:** AuthDBMAuthoritative On

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

**Invalidar:** AuthConfig

El valor por defecto de la directiva, permite que mod\_auth\_dbm se convierta en la autenticación completa para el directorio.

Por eso, si la autenticación basada en DBM falla para un usuario determinado, las credenciales del mismo no se pasan al esquema de autenticación de menor nivel.

Cuando asignamos el valor off, las credenciales de una autenticación fallida se pasan al siguiente nivel de autenticación.

**TRUCO:** Un uso habitual de este módulo es en unión con uno de los módulos básicos auth, como el mod\_auth.c. Mientras que este módulo DBM sustituye el conjunto de verificaciones de credenciales de usuario, unos cuantos (administrador) accesos relacionados caen directamente a un nivel inferior con el archivo .htpasswd bien protegido.

A continuación, vamos a ver un ejemplo de cómo puede utilizar un nombre de usuario y una contraseña basados en DBM. Suponiendo que tiene creado el archivo DBM de usuario, está capacitado para restringir el acceso a cualquier directorio Web. En el siguiente ejemplo, supongo que el archivo DBM de usuarios es /www/secrets/myuserdbm. Puede añadir el esquema de autentificación al servidor global o virtual utilizando un contenedor <Directory>, o puede utilizar el archivo .htaccess, no hay diferencia. El ejemplo de configuración es el siguiente:

```
AuthName "Apache Server Bible Readers Only"
AuthType Basic
AuthUserDBMFile /www/secrets/myuserdbm
require valid-user
```

Ahora Apache utiliza el módulo mod\_auth\_db para la autentificación en el directorio en el que se aplica la configuración.

**ADVERTENCIA:** Asegúrese de que sólo pueden leer el archivo DBM Apache y el dueño. Nadie excepto el dueño puede ser capaz de escribir en él.

## mod\_auth\_db

Si su sistema no es capaz de utilizar DBM, pero está disponible el soporte de Berkeley DB, puede utilizar mod\_auth\_db para utilizar archivos DB en lugar de los módulos DBM de Apache. Este módulo no está compilado en la distribución estándar de Apache.

Antes de configurar Apache con el módulo de autentificación basado en archivos DB, asegúrese de que sabe dónde están almacenados los archivos DB en su sistema. Por ejemplo, en un sistema Linux, los archivos se encuentran en el directorio estándar /usr/lib. Si su sistema no tiene las bibliotecas DB, tendrá que obtener el código fuente y compilar primero el soporte DB. Puede encontrar información sobre las bibliotecas DB en [www.sleepycat.com](http://www.sleepycat.com).

Una vez que se ha asegurado que su sistema tiene bibliotecas DB, puede proceder a la reconfiguración y la recompilación de Apache. Utilice la opción --enable-db con el script configure para configurar la fuente Apache para

el soporte de Berkeley DB, y entonces compile e instale Apache de la forma habitual.

En este momento, está listo para utilizar el módulo `mod_auth_db`. Este módulo `mod_auth_db` proporciona las directivas `AuthDBUserFile`, `AuthDBGroupFile` y `AuthDBAuthoritative`.

## **AuthDBUserFile**

La directiva `AuthDBUserFile` asigna el fully qualified pathname (nombre completo de la ruta de una máquina incluido su dominio) del usuario del archivo DB que contiene la lista de usuarios y las contraseñas encriptadas.

**Sintaxis:** `AuthDBUserFile nombre archivo`

**Contexto:** Directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** `AuthConfig`

Al igual que su equivalente DBM, el archivo de usuario DB también tiene como clave el nombre de usuario y el valor es la contraseña encriptada `crypt()`.

**ADVERTENCIA:** Asegúrese siempre de que sus archivos de usuario están guardados fuera del árbol de documentos Web y solo lo puede hacer Apache. Nadie, excepto el dueño (Apache) debería tener acceso a estos archivos.

## **AuthDBGroupFile**

La directiva `AuthDBGroupFile` asigna el fully qualified pathname (nombre completo de la ruta de una máquina incluido su dominio) del archivo de grupo DB, que contiene la lista de grupos de usuarios para la autenticación de usuarios. Al igual que su DBM equivalente, el archivo de grupo utiliza el nombre de usuario como llave y la lista de grupos separados por comas como valor. No debe haber espacios entre el valor, y nunca puede contener los dos puntos.

**Sintaxis:** `AuthDBGroupFile nombre archivo`

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** `AuthConfig`

Si prefiere no utilizar un archivo de grupos separados, puede utilizar un solo archivo DB para proporcionar la información de contraseña y de grupo. El formato de este archivo es:

```
DB_File_Key{username} = encrypted_password:  
comma_separated_group_list
```

donde el nombre de usuario (username) es la clave, y la contraseña y la lista de grupos son dos campos del valor. Se pueden dejar otros datos en el archivo DB tras dos puntos; el módulo de autenticación los ignorará. Si utiliza un solo DB para proporcionar la información del grupo y de la contraseña, tendrá que colocar las directivas AuthDBGroup y AuthDBUserFile en el mismo archivo.

## AuthDBAuthoritative

Cuando utiliza varios esquemas de autenticación como mod\_db, mod\_dbm y el estándar mod\_auth en el mismo directorio, puede utilizar la directiva AuthDBAuthoritative para determinar si mod\_auth\_db es el esquema de autenticación completo.

El valor por defecto de la directiva permite que mod\_auth\_db se convierta en la autenticación completa para el directorio. Por eso, si la autenticación basada en DB falla para un usuario determinado, las credenciales de este usuario no se pasan al esquema de autenticación de nivel inferior. Cuando se asigna el valor Off, las credenciales de una autenticación fallida se pasan al siguiente nivel de autenticación.

**Sintaxis:** AuthDBAuthoritative On | Off

**Predefinido:** AuthDBAuthoritative On

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

**Invalidar:** AuthConfig

## Módulos de generación de contenido dinámico

Los módulos que se discuten aquí permiten a Apache ejecutar scripts CGI, SSI, filtros y similares. La tabla 5.4 contiene una lista de estos módulos.

**Tabla 5.4.** Módulos de generación de contenido dinámico

Módulo	Función
mod_cgi	Ejecuta scripts CGI .
mod_include	Filtro SSI.
mod_actions	Ejecuta scripts CGI basados en los tipos MIME o en el método de la solicitud.
mod_ext_filter	Filtrar salidas con programas externos.

## **mod\_actions**

El módulo `mod_actions` está compilado por defecto. `mod_actions` le permite ejecutar un script de Perl basado en el tipo MIME o en el método de la solicitud HTTP. Ofrece las directivas siguientes.

### **Action**

La directiva `Action` le permite asociar una acción a un tipo MIME determinado. La acción es normalmente un script CGI que procesa el archivo que se ha solicitado. Esto le permite ejecutar un script CGI para un tipo MIME dado.

**Sintaxis:** `Action MIME_type cgi_script`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** `FileInfo`

Por ejemplo, la siguiente directiva hace que Apache ejecute el script específico cada vez que se solicita un archivo HTML:

```
Action text/html /cgi-bin/somescript.pl
```

El script recibe la URL y la ruta del archivo del documento solicitado mediante el CGI `PATH_INFO` estándar y las variables de entorno `PATH_TRANSLATED`. Esto puede ser útil en el desarrollo de los scripts de filtro. Esta sección discute uno de estos scripts de filtro. Cuando se solicita un archivo de texto (`.txt`) mediante la Web, aparece en el navegador Web en un formato muy poco deseable, porque la mayoría de los navegadores no pueden traducir los saltos de línea. Normalmente, la mayor parte de los archivos de texto aparecen como grandes párrafos. Utilizando la directiva `Action`, puede encontrar una solución mucho mejor. Para tener un ejemplo más interesante, vamos a imaginar que quiere desarrollar una solución que no sólo muestre mejor el archivo de texto en el navegador Web sino que, además, inserte un mensaje copyright al final de cada archivo de texto. Para llevar esto a cabo, necesita hacer dos cosas. Primero, añadir la siguiente directiva en el archivo `httpd.conf`:

```
Action plain/text /cgi-bin/textfilter.pl
```

Entonces, desarrollar el script Perl `textfilter` que mostrará el archivo de texto según sus preferencias. El listado 5.1 muestra un script de este tipo. Puede encontrar este script en el CD-ROM.

#### **Listado 5.1. `textfilter.pl`**

```
#!/usr/bin/perl  
#  
# Script: textfilter.pl
```

```

#
# Función: Este script de filtro convierte archivos de texto
#           en un documento HTML pero mantiene la composición del
#           texto.
#
# Copyright (c) 2001 by Mohammed J. Kabir
#
# Licencia: GPL
#
# El archivo del mensaje copyright se almacena siempre en
# el directorio raíz de documentos del servidor
# y se llama copyright.html.
#
my $copyright_file = $ENV{DOCUMENT_ROOT} . "/copyright.html";
# Obtiene la ruta del documento solicitado
my $path_translated = $ENV{PATH_TRANSLATED};

# Otras variables necesarias para almacenar datos
my $line;
my @text;
my @html;

# Almacena la información de la ruta y el nombre del archivo del
# documento solicitado en un array
@filename = split(/\//,$path_translated);

# Como se utilizan etiquetas HTML para mostrar el archivo de
# texto,
# vamos a imprimir la cabecera de contenido text/html.
print "Content-type: text/html\n\n";

# Lee el documento solicitado y almacena los datos
# en la variable array @text
@text = &readFile($path_translated);

# Ahora imprime las siguientes etiquetas de documento HTML.
# Estas etiquetas se enviarán antes que el contenido del
# verdadero documento
#
print <<HEAD;
<HTML>
<HEAD> <TITLE>$filename[-1] </TITLE> </HEAD>
<BODY BGCOLOR="white">
<BLOCKQUOTE>
<PRE>
HEAD

# Ahora imprime cada línea almacenada en el array @text
# (es decir, el contenido del documento solicitado)
#
foreach $line (@text) { print $line; }
# Ahora lee el archivo copyright y almacena el contenido

```

```

# en la variable array @html
#
@html = &readFile($copyright_file);
# Imprime cada línea almacenada en el array @html (es decir,
# el contenido del archivo de mensaje copyright)
#
foreach $line (@html){ print $line; }
# Sale del filtro
exit 0;

sub readFile {
#
# Subrutina: readFile
# Función: Lee un archivo si éste existe o sino imprime
# un mensaje de error y sale del script
#
# Obtiene el nombre del nombre del archivo pasado y lo guarda
# en la variable $file
my $file = shift;

# Variable local buffer
my @buffer;

# Si existe el archivo, lo abre y lee todas
# las líneas de la variable array @buffer
if(-e $file) {

    open(FP,$file) || die "Can not open $file.";
    while(<FP>){
        push(@buffer,$_);
    }

    close(FP);
} else {
    push(@buffer,"$file is missing.");
}

# Devuelve el contenido del buffer.
return (@buffer);
}

```

El script anterior lee el archivo de texto solicitado e imprime el contenido dentro de unas cuantas etiquetas HTML que permiten que se muestre el contenido tal y como es. Esto se realiza utilizando la etiqueta HTML <PRE>. Una vez que se ha impreso el contenido, el contenido del archivo del mensaje copyright se inserta al final de la salida. Esto permite que se imprima un mensaje copyright con cada archivo de texto solicitado.

La figura 5.1 muestra un ejemplo de salida en la que se muestra un archivo de texto en el navegador Web.

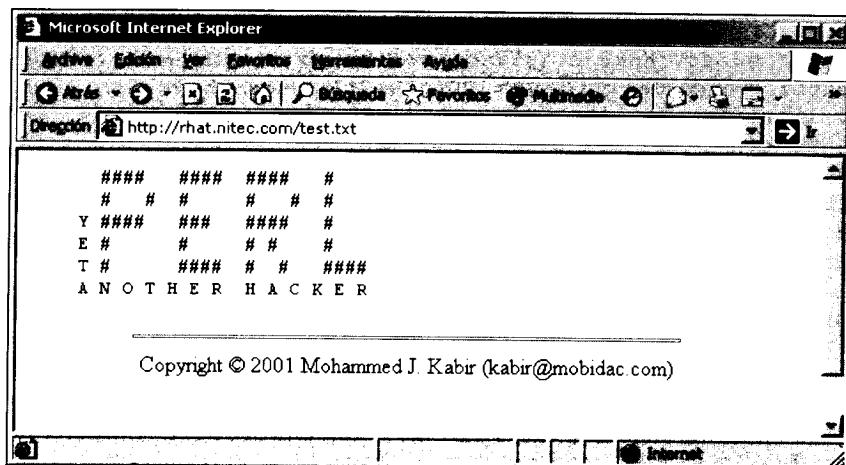


Figura 5.1. Salida del textfilter.pl

Como puede ver, el nombre del archivo solicitado, aparece como el título. El documento se encuentra entre comillas, y se imprime un mensaje de texto personalizado. El archivo del mensaje copyright se almacena en el directorio raíz de documentos. El archivo utilizado en este ejemplo es:

```
</PRE>
<BLOCKQUOTE>
<CENTER>
<HR>
Copyright © 2001 Mohammed J. Kabir (kabir@mobidac.com})
</CENTER>
</BODY>
</HTML>
```

## Script

La directiva **Script** es como la directiva **Action**, pero en lugar de asociar una acción con el tipo MIME, lo asocia con una solicitud HTTP como GET, POST, PUT o DELETE. El script CGI recibe la URL y la ruta del archivo del documento solicitado utilizando las variables de entorno estándar CGI PATH\_INFO y PATH\_TRANSLATED.

**Sintaxis:** Script method cgi-script

**Contexto:** configuración del servidor, host virtual, directorio

Esta directiva define la acción por defecto. En otras palabras, si tiene definido lo siguiente:

```
Script POST /cgi-bin/default_post.pl
```

en un archivo de configuración Apache (como `srm.conf`), entonces, cada vez que se realiza una solicitud mediante el método POST HTTP, será procesado

como es habitual, a no ser que tengamos que utilizar la acción por defecto especificada por la directiva.

```
<FORM METHOD="POST">
Enter Name: <INPUT TYPE=TEXT NAME="name" SIZE=25>
<INPUT TYPE=SUBMIT VALUE="Click Here">
</FORM>
```

Si un usuario envía un nombre mediante este formulario, no hay un script CGI especificado para procesar la información, por lo que en este caso, se ejecutará por defecto la acción POST del script /cgi-bin/default\_post.pl. Sin embargo, si se cambia la etiqueta <FORM . . .> a:

```
<FORM ACTION="/cgi-bin/form_processor.pl" METHOD="POST">
```

entonces cada vez que se envía el formulario, se llama al script /cgi-bin/form\_processor.pl como es habitual. Lo que haga en el script de la acción por defecto es decisión suya. En un sistema de un proveedor de Internet recomendando hacer que el script por defecto imprima mensajes significativos, para que el usuario del formulario HTML obtenga alguna pista de lo que está haciendo mal.

En caso de una solicitud GET, la acción por defecto se utiliza únicamente si la solicitud acompaña datos de consulta. Por ejemplo, www.yoursite.com/somefile.html se procesa de la forma habitual, pero si se recibe una solicitud como http://www.yoursite.com/somefile.html?some=data, se ejecutará la acción por defecto para GET.

## **mod\_ext\_filter**

El módulo mod\_ext\_filter permite a Apache utilizar un programa externo como filtro de entradas y salidas (input y output). Se puede utilizar cualquier programa que pueda leer una entrada de STDIN y cualquiera que pueda escribir una salida en STDOUT. Por supuesto, ejecutar un programa externo para procesar una entrada o una salida para cada solicitud es una tarea que consume tiempo y debe evitarse en un entorno de producción. Los filtros se desarrollan mejor utilizando el API (interfaz de aplicación de programas) de Apache y ejecutando dentro del proceso del servidor Apache.

### **ExtFilterDefine**

La directiva ExtFilterDefine le permite definir un filtro que el nombre del filtro especificado en esta directiva puede utilizar más tarde.

**Sintaxis:** ExtFilterDefine filter\_name [mode=input | output] [intype=MIME-type] [outtype=MIME-type] [PreservesContentLength]

**Contexto:** configuración del servidor

En el siguiente ejemplo, se define un filtro llamado `gzfilter` para que sea el filtro de salida (`mode=output`), que se ejecuta en el programa `/usr/bin/gzip` cuando se le llama:

```
ExtFilterDefine gzfilter mode=output cmd=/usr/bin/gzip
```

Aquí, `mode` sólo puede tener el valor `output`. Las asignaciones `intype` y `outtype` se utilizan para definir el tipo MIME utilizado para la salida y la entrada. Por ejemplo, si un filtro recibe `intype=text/plain` y `outtype=text/html`, entonces el filtro es responsable de traducir el dato del formato texto al formato HTML. El parámetro `PreservesContentLength` se podría utilizar cuando el filtro no cambie el tamaño de los datos en bytes.

Imagine que tiene su propio programa de filtrado llamado `/usr/local/bin/program` y quiere utilizarlo como un filtro `output` para archivos de texto en un directorio Web llamado `/www/mysite/htdocs/mytxts`. A continuación tenemos una muestra de la configuración que le permite realizarlo:

```
ExtFilterDefine my_test_filter \
    mode=output cmd=/usr/local/bin/program \
    intype=text/plain \
    outtype=text/html

<Directory "/www/mysite/htdocs/mytxts">

    SetOutputFilter my_test_filter
    AddType text/html

</Directory>
```

En este ejemplo, `ExtFilterDefine` define un filtro llamado `my_filter` que ejecuta `/usr/local/bin/program` cuando es llamado, y toma datos todo texto de `STDIN` y escribe `texto/html` en `STDOUT`. Ahora el contenedor `<Directory>` asigna este filtro como el filtro `output` para este directorio y además, le dice a Apache que el tipo `output` MIME es `texto/html` utilizando la directiva `AddType`. Si almacena archivos de texto en este directorio y los clientes Web lo solicitan, los archivos se traducirán a HTML utilizando el `/usr/local/bin/program`.

## ExtFilterOptions

La directiva `ExtFilterOptions` determina opciones de depuración de errores para el módulo. Cuando se necesita asignar una depuración de errores, utiliza la opción `DebugLevel`.

Fijar esta opción en 0, que es el valor por defecto, desactiva la depuración de errores. Fijar esta opción en 1 permite las entradas de registro de depuración de errores que muestran las opciones. Fijar esta opción en 9 permite todos y cada uno de los detalles del proceso de filtrado.

**Sintaxis:** ExtFilterOptions DebugLevel=n LogStderr |  
NoLogStderr

**Contexto:** servidor

## Módulos de configuración de tipo de contenido

Los módulos de esta sección, mostrados en la tabla 5.5, le permiten configurar, detectar y negociar tipos de contenido de la forma apropiada para servir solicitudes.

Tabla 5.5. Módulos Content-Type

Módulo	Función
mod_mime	Permite a Apache determinar el tipo MIME utilizando la extensión del archivo.
mod_mime_magic	Permite a Apache determinar el tipo MIME utilizando los números mágicos (patrones de bytes).
mod_negotiation	Permite a Apache realizar negociación de contenido enviando el mejor tipo de contenido que el cliente puede aceptar.

### mod\_mime

El módulo mod\_mime está compilado por defecto en Apache. Proporciona clientes con meta información sobre documentos. También le permite definir un manejador para un documento para determinar el modo en el que Apache procesa el documento.

#### AddCharset

La directiva AddCharset integra una o más extensiones de archivos a un carácter MIME fijo. Esto le permite asociar un carácter fijo a una o más extensiones de archivo.

**Sintaxis:** AddCharset charset file\_extension  
[file\_extension ...]

**Contexto:** configuración del servidor, host virtual, directorio, configuración en el ámbito de directorios (.htaccess)

**Invalidar:** FileInfo

El siguiente ejemplo da lugar a que un archivo llamado `filename.utf8` se integre como un carácter fijo llamado UTF-8.

```
AddCharset UTF-8 .utf8
```

## AddEncoding

La directiva `AddEncoding` integra una o más extensiones a un esquema de codificación MIME.

En otras palabras, esta directiva asocia un esquema de codificación a una o más extensiones de archivo.

**Sintaxis:** `AddEncoding MIME file_extension [file_extension...]`

**Contexto:** configuración del servidor, host virtual, directorio, configuración en el ámbito de directorios (`.htaccess`)

**Invalidar:** `FileInfo`

Por ejemplo, las siguientes directivas dan lugar a que un archivo llamado `backup.gz` se integre como un archivo `x-gzip-encoded`, y a que un archivo llamado `tarball.tar` se integre como un archivo `x-tar-encoded`.

```
AddEncoding x-gzip gz  
AddEncoding x-tar tar
```

## AddHandler

La directiva `AddHandler` define un manejador para una o más extensiones de archivo. Cada vez que Apache encuentra un archivo con un manejador definido, permite que el manejador procese el archivo.

**Sintaxis:** `AddHandler handler-name file-extension [file-extension ...]`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

El siguiente ejemplo, la directiva determina que todos los archivos `.cgi` se procesen con un manejador llamado `cgi-script`.

```
AddHandler cgi-script .cgi
```

## AddLanguage

La directiva `AddLanguage` integra una lista de extensiones de archivo a un idioma MIME. Cuando Apache encuentra un archivo con esa extensión sabe qué idioma soporta el archivo.

**Sintaxis:** `AddLanguage MIME_language file_extension [file_extension] [...]`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios ( .htaccess )

### **Invalidar:** FileInfo

El siguiente ejemplo integra todos los archivos con extensiones .en o .english para ser integrados como archivos English-language. Esto es útil en negociación de contenido, en donde el servidor puede devolver un documento basado en la preferencia de idioma del cliente.

```
AddLanguage en .en .english
```

O, en el ejemplo siguiente, si el cliente prefiere un documento en inglés y están disponibles tanto document.fr.html como document.en.html, el servidor debería devolver el documento document.en.html.

```
AddLanguage en .en  
AddLanguage fr .fr
```

## **AddType**

La directiva AddType integra una lista de extensiones de archivo al tipo MIME de modo que cuando Apache encuentra archivos con esas extensiones sabe qué tipo MIME utilizar.

**Sintaxis:** AddType MIME file\_extension [file\_extension ... ]

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios ( .htaccess )

### **Invalidar:** FileInfo

Por ejemplo, la siguiente línea asocia el tipo MIME llamado text/html a las extensiones htm, html, HTM y HTML.

```
AddType text/html htm html HTM HTML
```

## **DefaultLanguage**

La directiva DefaultLanguage asigna el idioma por defecto.

**Sintaxis:** DefaultLanguage MIME\_language

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios ( .htaccess )

### **Invalidar:** FileInfo

Por ejemplo, en la siguiente directiva todos los contenidos del directorio / www/mysite/Japanese están integrados al idioma por defecto, japonés:

```
<Directory /www/mysite/japanese>
    DefaultLanguage .jp
</Directory>
```

## ForceType

La directiva ForceType fuerza un tipo determinado de MIME para todos los archivos en un directorio.

El directorio se puede especificar con un contenedor <Directory> o un contenedor <Location>.

**Sintaxis:** ForceType MIME\_type

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

Por ejemplo, la siguiente directiva fuerza el tipo MIME text/html para todos los archivos en el directorio especificado, independientemente de sus extensiones:

```
<Directory /www/nitec/public/htdocs/files/with/no/extensions>
    ForceType text/html
</Directory>
```

## SetHandler

La directiva SetHandler define un manejador para un directorio o una localización URL. El manejador se utiliza para procesar todos los archivos en el directorio.

**Sintaxis:** SetHandler handler\_name

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

Por ejemplo, la siguiente directiva fuerza a que todos los archivos de la localización /bin sean tratados como scripts CGI, que son manejados por el manejador cgi-bin:

```
<Location /bin>
    Options ExecCGI
    SetHandler cgi-bin
</Location>
```

## RemoveHandler

La directiva RemoveHandler deshace un manejador para un directorio o para una localización URL. Es útil para limitar SetHandler, que normalmente se aplica a todos los archivos en el directorio.

Utilizando RemoveHandler puede eliminar manejadores para algunos archivos o, incluso, un subdirectorío.

**Sintaxis:** RemoveHandler handler\_name

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

Por ejemplo, en la siguiente directiva, el manejador my-handler se fija con una extensión .mjk fuera del directorio /www/mysite/htdocs/special, por lo que, automáticamente, se aplica también a este directorio. Sin embargo, como RemoveHandler se aplica a este directorio para deshacer la asociación entre my-handler y .mjk, los archivos con extensiones .mjk en este directorio no se manejan con my-handler:

```
SetHandler my-handler .mjk

<Directory /www/mysite/htdocs/special>
    RemoveHandler .mjk
</Location>
```

## TypesConfig

La directiva TypesConfig determina el archivo de configuración MIME por defecto. El valor por defecto debería ser adecuado para la mayoría de las instalaciones Apache. Si quiere añadir sus propios tipos MIME, utilice la directiva AddType en lugar de modificar este archivo.

**Sintaxis:** TypesConfig nombre archivo

**Predefinido:** TypesConfig conf/mime.types

**Contexto:** configuración del servidor

**NOTA:** Si necesita soporte adicional para manejar tipos MIME, tendrá que dirigirse al módulo mod\_mime\_magic de la sección siguiente. Para la mayor parte de las instalaciones Apache esto no es necesario, por lo que no se discute en este libro.

## mod\_mime\_magic

El módulo mod\_mime\_magic le permite a Apache determinar el tipo de un archivo MIME comparando unos cuantos bytes del archivo con un valor mágico almacenado en un archivo. Este módulo sólo se necesita cuando mod\_mime no puede adivinar el tipo MIME de un archivo. En la mayoría de los casos, no necesita este módulo. Este módulo tiene una directiva llamada MimeMagicFile.

Esta directiva permite el módulo mod\_mime\_magic y señala el archivo mágico necesario para este módulo. La distribución Apache contiene un archivo mágico en el subdirectorio conf, por lo que si quiere utilizar este módulo, fije esta directiva en conf/magic.

**Sintaxis:** MimeMagicFile magic\_file\_filename

**Contexto:** configuración del servidor, host virtual

## mod\_negotiation

El módulo mod\_negotiation está compilado por defecto. Proporciona soporte para negociaciones de contenido. En un escenario típico de negociación de contenido, el cliente proporciona información sobre el tipo de contenido que puede manejar, y el servidor intenta proporcionar el contenido más apropiado. El servidor lleva esto a cabo con la ayuda de integración de tipos y el MultiViews.

Un tipo map proporciona una descripción de documentos. Cada descripción de documentos contiene una o más cabeceras. Puede contener también líneas de comentarios que comienzan con un carácter almohadilla (#). Las descripciones de documentos están separadas por líneas en blanco. Las cabeceras de descripción de documentos son:

- **Content-Encoding:** especifica el tipo de codificación del archivo. Únicamente están permitidas la codificación x-compress y x-gzip.
- **Content-Language:** el idioma del documento.
- **Content-Length:** la longitud del archivo en bytes.
- **Content-Type:** el tipo MIME del documento. Están permitidos parámetros opcionales clave-valor. Los parámetros permitidos son level, que proporciona el número de versión (como un integer) del tipo MIME, y qs, que indica la cualidad (como un número decimal floating) del documento.
- **URI:** la ruta del documento referida al archivo de integración.

El buscador MultiViews trata de determinar la coincidencia más cercana para el documento perdido utilizando la información que tiene del cliente, y devuelve esta correspondencia si es posible. Cuando permite la opción MultiViews en la directiva Options, el servidor es capaz de realizar la búsqueda MultiViews cuando no se encuentra el documento solicitado. Este módulo proporciona las dos directivas siguientes.

### CacheNegotiatedDocs

La directiva CacheNegotiatedDocs permite que se almacenen en el caché los documentos de contenido negociado en servidor proxy. Tenga en cuenta que la nueva especificación HTTP 1.1 proporciona mucho más control para cachear los documentos negociados, y CacheNegotiatedDocs no tiene efecto sobre las respuestas a las solicitudes HTTP 1.1. Esta directiva ha desaparecido prácticamente desde la aparición de HTTP 1.1. No se recomienda la utilización de CacheNegotiatedDocs.

**Sintaxis:** CacheNegotiatedDocs

**Contexto:** configuración del servidor

## LanguagePriority

La directiva `LanguagePriority` determina el idioma que el servidor va a preferir en un escenario de búsqueda `MultiViews`, cuando el cliente no proporciona ninguna información sobre sus preferencias de idioma.

**Sintaxis:** `LanguagePriority MIME_language [MIME_language ...]`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** `FileInfo`

En la siguiente directiva, por ejemplo, si está activada la opción `MultiViews` y el cliente no proporciona ninguna información sobre sus preferencias de idioma para un archivo perdido, el servidor primero sirve la versión en inglés de la coincidencia más cercana, y sino la francesa, y así sucesivamente. Al igual que la directiva `CacheNegotiatedDocs`, esta directiva no es efectiva en el entorno HTTP 1.1.

`LanguagePriority en fr de`

## Módulos de listas de directorios

Si tiene un directorio dentro de su árbol de documentos Web que no tiene un archivo con el índice de directorios (asignado utilizando la directiva `DirectoryIndex`) entonces Apache generará automáticamente una lista de directorios, en el caso de que no haya desactivado la lista de directorios automática utilizando la directiva `Options -Indexes`. Apache le permite personalizar la lista de directorios generada automáticamente. Los módulos de esta sección, que se muestran en la tabla 5.6, le permiten configurar el modo en el que se muestran las listas de directorios.

**Tabla 5.6.** Módulos de listas de directorios

Módulo	Función
<code>mod_dir</code>	Manejador básico de directorios.
<code>mod_autoindex</code>	Lista automática de directorios.

## **mod\_dir**

El módulo mod\_dir está compilado en Apache por defecto. Utilizando este módulo, Apache puede redirigir cualquier solicitud que no incluya ninguna barra final. Por ejemplo, este módulo puede redirigir `www.yoursite.com/somedirectory` a `www.yoursite.com/somedirectory/`. Además proporciona la directiva DirectoryIndex para ayudar con la realización de un índice del contenido del directorio.

La directiva DirectoryIndex especifica el nombre(s) de los archivos que Apache debería buscar antes de crear un índice dinámico de directorios. Los archivos pueden ser de cualquier tipo, desde un archivo HTML hasta un script CGI. El valor por defecto permite a Apache buscar el archivo `index.html` para cualquier solicitud que termine con el nombre de un directorio.

**Sintaxis:** `DirectoryIndex local_URL [local_URL ...]`

**Predefinido:** `DirectoryIndex index.html`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** `Indexes`

Por ejemplo, `www.yoursite.com/some/directory/` hace que Apache busque un archivo llamado `/some/directory/index.html`. Si el archivo existe; su contenido es enviado al cliente. En ausencia de este archivo, Apache crea una lista dinámica de directorios.

Puede especificar por defecto uno o más archivos, como el archivo de índice de directorios. En el siguiente ejemplo, se le dice a Apache que busque todos los archivos nombrados para cada solicitud de un directorio:

```
DirectoryIndex index.html index.htm welcome.html welcome.htm
```

Tenga en cuenta que Apache buscará los archivos en el mismo orden (de izquierda a derecha) en el que aparecen en la configuración anterior. En otras palabras, si Apache encuentra `index.html`, no buscará `index.htm`, `welcome.html` o `welcome.htm`. Puede especificar un nombre de script CGI script como el índice por defecto. Por ejemplo, la siguiente directiva consigue que Apache ejecute el script `/cgi-bin/show_index.cgi` cada vez que Apache obtiene la solicitud de un directorio:

```
DirectoryIndex /cgi-bin/show_index.cgi
```

## **mod\_autoindex**

El módulo mod\_autoindex está compilado en Apache por defecto. Cuando Apache recibe una solicitud para un directorio, busca uno o más archivos de

índices de directorios especificados por la directiva `DirectoryIndex`. Normalmente este archivo es `index.html` o `index.htm`. En ausencia de ese archivo de índices, sin embargo, Apache puede generar una lista dinámica de directorios. Este módulo le permite controlar el modo en el que Apache crea la lista dinámica de directorios.

Apache genera dos tipos de índices dinámicos de directorios: simple y personalizada. El índice personalizado y otras muchas opciones de índices están disponibles para este módulo. Las directivas para `mod_authoindex` son las siguientes.

## AddAlt

Cuando está activada `FancyIndexing`, esta directiva determina el texto especificado como una alternativa al ícono que se muestra para uno o más archivos o extensiones de archivos especificados como argumentos. Esto se realiza para navegadores no gráficos como Linx.

**Sintaxis:** `AddAlt "text" nombrearchivo [nombrearchivo ...]`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** `Indexes`

Por ejemplo, la directiva siguiente le permite a Apache mostrar el texto alternativo "Pictures" en lugar del ícono para cada tipo de archivo gráfico especificado aquí. Para los navegadores gráficos como Netscape Navigator o Internet Explorer, el texto alternativo se muestra como texto de ayuda bajo las populares plataformas Windows. En estos sistemas, los usuarios pueden obtener un aviso o ayuda sobre el archivo cuando pasan el ratón por encima del ícono que representa a alguno de los tipos de archivo:

```
AddAlt "Pictures" gif jpeg jpg bmp
```

## AddAltByEncoding

Si no quiere asignar un texto alternativo a los nombres de archivo o a las extensiones de los archivos mediante la directiva `AddAlt`, puede utilizar la directiva `AddAltByEncoding` para asignar ese texto para una o más codificaciones MIME. Al igual que `AddAlt`, esta directiva sólo se puede utilizar cuando está activada `FancyIndexing`.

**Sintaxis:** `AddAltByEncoding "text" MIME_encoding [MIME_encoding ...]`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** `Indexes`

Por ejemplo, la siguiente directiva hace que Apache muestre el texto alternativo "Compressed File" ("Archivo comprimido") para todos los archivos de tipo MIME x-compress.

```
AddAltByEncoding "Compressed File" x-compress
```

## AddAltByType

Al igual que la directiva AddAltByEncoding, la directiva AddAltByType asigna texto alternativo para un archivo, en lugar de un ícono para FancyIndexing. Sin embargo, utiliza un tipo MIME en lugar de codificación MIME.

**Sintaxis:** AddAltByType "text" MIME-type [MIME\_type ...]

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

**Invalidar:** Indexes

Por ejemplo, la siguiente directiva muestra el texto "HTML FILE" en lugar del ícono, en los navegadores no gráficos. En el caso de navegadores gráficos, este texto aparecerá como aviso o ayuda:

```
AddAltByType "HTML FILE" text/html
```

## AddDescription

La directiva AddDescription asigna un texto descriptivo para un nombre de archivo, a un nombre de archivo parcial o a un nombre de archivo comodín cuando está activada FancyIndexing.

**Sintaxis:** AddDescription "text" file [ file ... ]

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

**Invalidar:** Indexes

Por ejemplo, la siguiente directiva muestra la descripción para todos los archivos GIF, JPEG, JPG y BMP generados en la lista de directorios:

```
AddDescription "Graphics File" *.gif *.jpeg *.jpg *.bmp
```

## AddIcon

La directiva AddIcon le permite asignar iconos a nombres de archivos y de directorios que se muestran para FancyIndexing.

**Sintaxis:** AddIcon icono nombre archivo [nombrearchivo ... ]

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

#### **Invalidar:** Indexes

Por ejemplo, la siguiente directiva le dice a Apache que muestre /icons/picture.gif cerca de los archivos que tengan extensiones .gif, .jpg y .bmp:

```
AddIcon /icons/picture.gif .gif .jpg .bmp
```

Si además, quiere proporcionar texto alternativo para la extensión de los archivos de la lista, puede utilizar un formato del siguiente tipo, en donde IMG es el texto alternativo que se muestra en los navegadores no gráficos:

```
AddIcon (IMG, /icons/picture.gif) .gif .jpg .bmp
```

Si quiere mostrar un ícono para un directorio, puede utilizar la directiva del siguiente modo:

```
AddIcon /path/to/your/directory/icon      ^^DIRECTORY^^
```

De igual manera, si quiere mostrar un ícono para cada línea en blanco mostrada por el esquema de índice personalizada, puede utilizar:

```
AddIcon /path/to/your/blank/line/icon      ^^BLANKICON^^
```

### **AddIconByEncoding**

La directiva AddIconByEncoding le deja asignar íconos a las codificaciones MIME. En otras palabras, puede asignar una imagen de un ícono a un tipo MIME.

**Sintaxis:** AddIconByEncoding icon\_file MIME\_encoding [MIME\_encoding...]

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

#### **Invalidar:** Indexes

Por ejemplo, la siguiente directiva AddIconByEncoding le dice a Apache que muestre el ícono /icons/zip.gif en todos los archivos que sean del tipo MIME x-gzip (por ejemplo los que tienen la extensión .gz).

```
AddIconByEncoding /icons/zip.gif          x-gzip
```

### **AddIconByType**

La directiva AddIconByType también le permite asignar íconos a uno o más tipos MIME.

**Sintaxis:** AddIconByType icon\_file MIME\_type [MIME\_type...]

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

**Invalidar:** Indexes

Por ejemplo, la siguiente directiva AddIconByType le dice a Apache que muestre /icons/html.gif para todos los archivos texto/html.

```
AddIconByType (HTML,/icons/html.gif) text/html
```

## DefaultIcon

Cuando AddIcon, AddIconByEncoding o AddIconByType no encuentran asociación para un determinado archivo, se puede mostrar un ícono por defecto. La directiva DefaultIcon le permite asignar ese ícono.

**Sintaxis:** DefaultIcon URL

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

**Invalidar:** Indexes

Por ejemplo, la siguiente directiva muestra idontknow.gif como ícono de cualquier archivo cuya asociación sea desconocida:

```
DefaultIcon /icon/idontknow.gif
```

## FancyIndexing

La directiva FancyIndexing le permite activar y desactivar un índice personalizado de directorios. Puede obtener el mismo efecto con la directiva IndexOptions.

**Sintaxis:** FancyIndexing On | Off

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

**Invalidar:** Indexes

## HeaderName

Si utiliza FancyIndexing, puede insertar contenido de un archivo al principio de la lista de índices. La directiva HeaderName le permite especificar el nombre del archivo para tal inserción.

**Sintaxis:** HeaderName nombre archivo

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

**Invalidar:** Indexes

Por ejemplo, la siguiente directiva le dice a Apache que busque un archivo llamado `welcome` o `welcome.html` en el directorio de la lista; si se encuentra ese archivo, se inserta el contenido antes de la lista real:

```
HeaderName welcome
```

## IndexIgnore

Si necesita que sean visibles algunos archivos o extensiones de archivos en la lista de directorios, puede utilizar la directiva `IndexIgnore`.

**Sintaxis:** `IndexIgnore file [file...]`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** `Indexes`

Por ejemplo, la siguiente directiva garantiza que Apache no mete en la lista de directorios los archivos `welcome`, `welcome.html` o los archivos de configuración del ámbito de directorios (`.htaccess`):

```
IndexIgnore welcome welcome.html .htaccess
```

El carácter `.` (punto) se encuentra automáticamente en la lista `IndexIgnore`; por eso, los archivos que comienzan por este carácter no se encuentran en la lista. Sin embargo, puede preferir añadir configuración del ámbito de directorios (`.htaccess`) en la lista, para sentirse más seguro.

## IndexOptions

La directiva `IndexOptions` especifica el comportamiento de la generación automática de un índice de directorios.

**Sintaxis:** `IndexOptions option [option] [...]`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** `Indexes`

La tabla 5.7 muestra las opciones que puede utilizar con `IndexOptions`.

**Tabla 5.7.** Opciones para `IndexOptions`

Opción	Lo que hace
<code>FancyIndexing</code>	Activa los índices de directorios personalizados. Tenga en cuenta que las directivas <code>FancyIndexing</code> e <code>IndexOptions</code> se invalidan la una a la otra.

Opción	Lo que hace
IconHeight [=pixels]	Permite a Apache incluir el atributo HEIGHT= pixels en la etiqueta IMG del ícono, lo que hace que la carga del ícono sea más rápida en la mayoría de los navegadores. Si no especifica el tamaño en píxeles, se utiliza un estándar por defecto.
IconsAreLinks	Hace que los iconos formen parte del anchor para el nombre de archivo, en los índices personalizados.
IconWidth [=pixels]	Permite a Apache incluir el atributo WIDTH= pixels en la etiqueta IMG del ícono, lo que hace que la carga del ícono sea más rápida en la mayoría de los navegadores. Si no especifica el tamaño en píxeles, se utiliza un estándar por defecto.
ScanHTMLTitles	Si quiere que Apache lea el título (indicado por el par de etiquetas <TITLE> y </TITLE>) de un documento HTML para un índice personalizado, utilice esta opción. Si tiene ya especificada una descripción utilizando la directiva AddDescription, sin embargo, esta opción no se utiliza. Tenga en cuenta que leer cada contenido de los archivos y buscar el título es una tarea que gasta mucho tiempo y que hará que se ralentice el envío de la lista de directorios. No recomiendo esta opción.
SuppressColumnSorting	Por defecto, Apache permite pinchar en los encabezados de las columnas de los índices de directorios personalizados, lo que permite a los usuarios ordenar información en esa columna. Esta opción inactiva esta característica.
SuppressDescription	Si no quiere mostrar descriptores de archivos en la lista de directorios personalizada, utilice esta opción.
SuppressHTMLPreamble	Si el directorio realmente contiene un archivo especificado por la directiva HeaderName, el módulo normalmente incluye el contenido del archivo después de un preámbulo HTML estándar (<HTML>, <HEAD>, etc.). La opción SuppressHTMLPreamble desactiva este comportamiento.

Opción	Lo que hace
SuppressLastModified	Suprime el despliegue de la última fecha de modificación en las listas de directorios personalizada.
SuppressSize	Suprime el tamaño del archivo en una lista de índices personalizada.

## IndexOrderDefault

La directiva `IndexOrderDefault` le permite cambiar la vista de la lista de directorios ordenando varios campos como nombre, fecha, tamaño y descripción en los directorios que se han mostrado utilizando la característica `FancyIndexing`.

**Sintaxis:** `IndexOrderDefault Ascending | Descending Name | Date | Size | Description`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** `Indexes`

## ReadmeName

Si quiere insertar un archivo al final de la lista de directorios personalizada, utilice la directiva `ReadmeName`.

**Sintaxis:** `ReadmeName nombrearchivo`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** `Indexes`

Por ejemplo, la siguiente directiva hace que Apache busque un archivo llamado `readme.html` o `readme` para insertarlo al final de la lista:

```
ReadmeName readme
```

## Response Header Modules

Apache le permite enviar cabeceras de respuestas HTTP al cliente cuando envía datos. Los módulos de esta sección, mostrados en la tabla 5.8, le permiten configurar varias cabeceras de respuesta.

**Tabla 5.8. Response-Header Modules**

Módulo	Función
mod_asis	Envía archivos que contiene sus propias cabeceras HTTP.
mod_headers	Añade cabeceras HTTP de forma arbitraria a los recursos.
mod_expires	Aplica Expires: cabeceras a los recursos.
mod_cern_meta	Soporte para meta archivos con cabeceras HTTP.

## **mod\_asis**

El módulo mod\_asis está compilado por defecto. Este módulo le permite enviar un documento tal cual es, en otras palabras, el documento se envía al cliente sin cabeceras HTTP. Esto puede ser útil cuando redirigimos clientes sin ayuda de ningún script. Para enviar un archivo tal cual es, necesita asegurarse de que el archivo httpd.conf contiene una entrada del siguiente tipo:

```
AddType httpd/send-as-is asis
```

Esto asigna el tipo MIME httpd/send-as-is a la extensión de archivo .asis. Si crea un archivo llamado foobar.asis y un cliente lo solicita, el archivo se envía al cliente sin ninguna cabecera HTTP. Es su trabajo incluir las cabeceras apropiadas en el archivo. Por ejemplo, si quiere proporcionar un mecanismo de redirección mediante los archivos .asis, puede crear archivos con cabeceras del tipo:

```
Status: 301 Text Message
Location: new-URL
Content-type: text/html
```

El listado 5.2 muestra un archivo llamado redirect.asis, que redirecciona al cliente a una nueva localización.

### **Listado 5.2. redirect.asis**

```
Status: 301 We have moved.
Location: http://www.our-new-site/
Content-type: text/html
<H1>Notice to Visitors</H1>
Please update your bookmark to point to <A href="http://
www.our-new-site/ "> www.our-new-site/ </A><br>
<br>
Thanks.
```

Cuando el cliente solicita este archivo, el mensaje de estado 301 le dice que utilice la información de localización para redireccionar la solicitud. No tiene que añadir las cabeceras `Date:` y `Server:`, porque el servidor las añade automáticamente. Sin embargo, el servidor no proporciona una cabecera `Last-Modified`.

## mod\_headers

Este módulo no está compilado por defecto. `mod_headers` le permite manipular las cabeceras de respuesta HTTP, y proporciona una sola directiva llamada `Header`, que le permite manipular la cabecera de respuesta HTTP.

**Sintaxis:** `Header action header value`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** `FileInfo`

Las acciones permitidas son:

Acción	Lo que hace
Set	Fija una cabecera. Si existe una cabecera antigua con el mismo nombre, su valor se cambia por uno nuevo.
Add	Añade una cabecera. Puede dar lugar a varias cabeceras con el mismo nombre cuando existen una o más cabeceras con el mismo nombre.
Append	Adjunta el valor de una cabecera que ya existe.
Unset	Elimina la cabecera.

Por ejemplo, la siguiente directiva añade la cabecera `Author` con el valor "Mohammed J. Kabir":

```
Header add Author "Mohammed J. Kabir"
```

Y la siguiente línea elimina la misma cabecera:

```
Header unset Author
```

## mod\_expires

El módulo `mod_expires` no está compilado en Apache por defecto. Le permite determinar el modo en el que Apache trata con cabeceras `Expires` HTTP en

la respuesta del servidor a las solicitudes. Las cabeceras Expires HTTP le proporcionan una forma de hablarle al cliente sobre el tiempo que tardan los recursos solicitados en inactivarse. Esto resulta útil cuando los documentos están en la memoria caché del cliente y ha de solicitarlos de nuevo. La mayor parte de los clientes determinan la validez de un documento solicitado investigando el tiempo de expiración de los documentos en el caché proporcionado por las cabeceras Expires HTTP. Este módulo le permite controlar las asignaciones de las cabezas Expires HTTP.

## **ExpiresActive**

La directiva ExpiresActive activa o inactiva la generación de la cabecera Expires. No garantiza que una cabecera Expires se genere. Si no se encuentra el criterio, no se envía la cabecera.

**Sintaxis:** ExpiresActive On | Off

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

**Invalidar:** Indexes

## **ExpiresByType**

La directiva ExpiresByType especifica el valor de la cabecera Expires HTTP para documentos de un tipo específico de MIME-type. El tiempo de expiración se determina en segundos. Puede definir el tiempo de dos formas. Si elige utilizar el formato Mseconds para marcar el tiempo de expiración, se utiliza el momento de la última modificación como base. En otras palabras, M3600 significa que quiere que el archivo expire una hora después de su modificación. Por otro lado, si utiliza el formato Aseconds, entonces el momento de acceso de clientes se utiliza como base de tiempo. A continuación tenemos algunos ejemplos.

**Sintaxis 1:** ExpiresByType MIME\_type Mseconds | Aseconds

**Sintaxis 2:** ExpiresByType MIME-type "base\_time [plus] num Years|Months|Weeks|Days|Hours|Minutes|Seconds"

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (.htaccess)

**Invalidar:** Indexes

Las siguientes directivas hacen que todos los archivos de texto expiren una hora después de estar en el caché del cliente:

```
ExpiresByType text/plain A3600
```

Y las siguientes consiguen que todos los archivos GIF expiren tras una semana de su última modificación:

```
ExpiresByType image/gif M604800
```

Si quiere utilizar la segunda sintaxis para especificar el tiempo de expiración, necesita determinar el valor apropiado de tiempo base utilizando las siguientes opciones:

Valor	Lo que significa
Access	Momento en el que el cliente accedió al archivo.
Now	Momento actual. Es lo mismo que el momento de acceso.
Modification	Momento en el que el archivo fue cambiado por última vez.

Por ejemplo, las siguientes directivas le dicen a Apache que envíe cabeceras para decirle al navegador que los documentos HTML expirarán tras siete días desde el momento de su acceso y que las imágenes GIF expirarán después de cualquier cambio en el archivo o tres horas y diez minutos después.

```
ExpiresByType text/html "access plus 7 days"  
ExpiresByType image/gif "modification plus 3 hours 10 minutes"
```

## ExpiresDefault

La directiva `ExpiresDefault` asigna el momento de expiración por defecto para todos los documentos en el contexto en que se encuentran especificados. Por ejemplo, si esta directiva está especificada en el host virtual, sólo se aplicará a los documentos accesibles mediante el host virtual. Del mismo modo, puede especificar esta directiva en un contexto en el ámbito de directorios, lo que permitirá que todos los documentos en ese directorio expiren en un intervalo específico. Ver `ExpiresByType` para obtener los detalles de sintaxis.

**Sintaxis 1:** `ExpiresDefault Mseconds | Aseconds`

**Sintaxis 2:** `ExpiresDefault "base_time [plus] num Years|Months|Weeks|Days|Hours|Minutes|Seconds"`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorios (`.htaccess`)

**Invalidar:** `Indexes`

A continuación tenemos dos ejemplos:

```
ExpiresDefault M3600  
ExpiresDefault "access plus 2 days"
```

El primer ejemplo marca el momento de expiración en una hora después de la última modificación de los documentos. El segundo marca el momento de expiración en dos días después de que el cliente acceda.

## **mod\_cern\_meta**

El módulo `mod_cern_meta` no está compilado por defecto. Proporciona soporte para meta información. Esta información puede consistir en cabeceras HTTP adicionales como:

```
Expires: Saturday, 19-May-01 12:00:00 GMT
```

o puede ser cualquier otro tipo de información como la que tenemos a continuación, en la que la meta información se almacena en un archivo y aparece junto con cabecera de la respuesta HTTP:

```
Foo=Bar
```

## **MetaFiles**

La directiva `MetaFiles` activa o desactiva el procesamiento de los archivos de meta cabecera.

**Sintaxis:** `MetaFiles On | Off`

**Predefinido:** `MetaFiles Off`

**Contexto:** archivo de control de acceso en el ámbito de directorio (`.htaccess`)

## **MetaDir**

La directiva `MetaDir` especifica el nombre del directorio que almacena los archivos meta cabecera. Por ejemplo, si tiene un directorio llamado `/www/mycompany/public/htdocs` y quiere almacenar archivos meta cabecera para ese directorio, necesitaría crear un subdirectorio llamado `.web` si utiliza el valor por defecto de la directiva `MetaDir`. El directorio `.web` almacena archivos meta cabecera.

**Sintaxis:** `MetaDir directorio`

**Predefinido:** `MetaDir .web`

**Contexto:** archivo de control de acceso en el ámbito de directorio (`.htaccess`)

## **MetaSuffix**

La directiva `MetaSuffix` especifica la extensión del nombre de archivo para los archivos de meta información. Por ejemplo, si tiene un archivo HTML llamado `mypage.html`, entonces necesita crear `mypage.html.meta` (utilizando el valor por defecto de esta directiva) para almacenar sus meta cabeceras. El archivo `mypage.html.meta` debe residir en el directorio especificado por la directiva `MetaDir`.

**Sintaxis:** `MetaSuffix suffix`

**Predefinido:** `MetaSuffix .meta`

**Contexto:** archivo de control de acceso en el ámbito de directorio (`.htaccess`)

Para permitir a Apache que envíe meta información sobre un directorio llamado `/www/mycompany/public/htdocs`, necesita hacer lo siguiente:

1. Fijar la directiva `MetaFiles` en `on` en el archivo de configuración en el ámbito de directorio (`.htaccess`) para `/www/mycompany/public/htdocs`. También puede asignar las directivas `MetaDir` y `MetaSuffix` en este archivo.
2. Crear un subdirectorio llamado `.web` (suponiendo que está utilizando el valor por defecto de la directiva `MetaDir`).
3. Crear un archivo de texto con la extensión `.meta` (suponiendo que está utilizando el valor por defecto de la directiva `MetaSuffix`).
4. Poner todas las cabeceras HTTP que quiera suministrar en este archivo.

Por ejemplo, para proporcionar meta cabeceras para un archivo llamado `/www/mycompany/public/htdocs/mypage.html`, necesita crear un archivo llamado `/www/mycompany/public/htdocs/.web/mypage.html.meta`. Este archivo puede incluir líneas como estas:

```
Expires: Saturday, 19-May-01 12:00:00 GMT
Anything=Whatever
```

## **Módulos de información de servidores y de registro**

Los módulos de esta sección, mostrados en la tabla 5.9, le permiten registrar accesos, informar sobre el estado del servidor y dar información de configuración, e incluso, seguir la pista de usuarios que están utilizando cookies.

**Tabla 5.9.** Módulos de información de servidores y de registro

Módulo	Función
mod_log_config	Proporciona registros de acceso personalizados.
mod_status	Muestra información sobre el estado.
mod_info	Muestra información sobre la configuración del servidor.
mod_usertrack	Proporciona un seguimiento de clientes utilizando Cookies http.

## mod\_log\_config

Este módulo se discute en detalle en el capítulo 8. Ver la sección "Crear archivos de registro" en ese capítulo para aprender más cosas sobre este módulo y sus directivas.

## mod\_status

Este módulo se discute en detalle en el capítulo 8. Ver la sección "Permitir páginas de estado con mod\_status" en ese capítulo para aprender más cosas sobre este módulo y sus directivas.

## mod\_info

Este módulo se discute en detalle en el capítulo 8. Ver la sección "Acceder a la configuración de acceso con mod\_info" en ese capítulo para aprender más cosas sobre este módulo y sus directivas.

## mod\_usertrack

Este módulo se discute en detalle en el capítulo 8. Ver la sección "Registrar cookies" en ese capítulo para aprender más cosas sobre este módulo y sus directivas.

# Módulos de integración URL

Los módulos en esta sección, mostrados en la tabla 5.10, le permiten integrar distintas URL a directorios físicos determinados, crear reglas complejas de reescritura, crear alias y automatizar las URL de host virtuales a integraciones de directorios físicos.

**Tabla 5.10.** Módulos de integración de URL

Módulo	Función
mod_userdir	Le permite acceder a sitios Web personales almacenados en directorios locales del usuario.
mod_rewrite	La reglas de reescritura de URL se crean utilizando este módulo. Ver el capítulo 9 para obtener los detalles.
mod_alias	Para integrar distintas partes del sistema de archivos del host en el árbol de documentos y para redireccionar URL.
mod_speling	Corrección automática de pequeños errores en las URL.
mod_vhost_alias	Soporte de configuración dinámica en masa de alojamientos virtuales.

## **mod\_userdir**

El módulo `mod_userdir` le permite a Apache hacer accesibles directorios Web específicos de usuario mediante `http://your_server_name/~username`. Si no tiene pensado el soporte de ese tipo de sitios Web, no necesita este módulo. La directiva `UserDir` (la única directiva de este módulo) le permite asignar el directorio que Apache debería considerar como raíz de documentos para el sitio Web del usuario.

**Sintaxis:** `UserDir directorio_nombredirectorio`

**Predefinido:** `UserDir public_html`

**Contexto:** configuración del servidor, host virtual

Por ejemplo, si mantiene el valor por defecto, cada vez que Apache reconozca una ruta `~username` después del nombre del servidor en la URL solicitada, traducirá el `~username` a `user_home_directory/public_html`. Si los directorios locales del usuario están almacenados en `/home`, la ruta traducida es `/home/username/public_html`.

**ADVERTENCIA:** Debería añadir un `Userdir disabled root` para invalidar la capacidad de asignar esta directiva para señalar el directorio raíz.

El nombre del directorio que asigna con esta directiva debe ser accesible para el servidor Web. En otras palabras, si `/home/username` es el directorio local

deje Userdir asignado con el valor public\_html, entonces /home/username/public\_html debe ser accesible para el servidor Web. De hecho, Apache necesitará también leer y ejecutar acceso a ambos directorios, /home y /home/username.

Algunos administradores preocupados por la seguridad no son partidarios de esta idea de crear un directorio Web en un directorio local del usuario, puede asignar el UserDir a una ruta distinta como:

```
UserDir /www/users
```

Ahora, cuando se solicite `http://your_server_name/~username`, Apache traducirá esta solicitud a /www/users/username. De este modo puede mantener los archivos Web del usuario fuera del directorio local (/home/username) creando un nuevo directorio de nivel superior en el que tiene que crear un directorio para cada usuario.

Recuerde asegurarse de que Apache tiene acceso de lectura y escritura a cada uno de estos directorios.

## **mod\_alias**

El módulo mod\_alias está compilado por defecto en Apache. Proporciona varias directivas que se pueden utilizar para asociar una parte del sistema de archivos del servidor a otra o, incluso, realizar servicios de redirección de URL.

### **Alias**

La directiva Alias le permite asociar una ruta a cualquier sitio de su sistema de archivos del sistema.

**Sintaxis:** Alias URL-ruta ruta

**Contexto:** configuración del servidor, host virtual

Por ejemplo, la siguiente directiva asocia /data/ con /web/data; por lo tanto, cuando `http://www.yoursite.com/data/`:

```
Alias /data/ "/web/data/"
```

recibe una solicitud del tipo `http://data/datafile.csv`, se devuelve el archivo llamado /web/data/datafile.csv.

Es importante recordar que las rutas deben ser relativos dentro de su sistema de archivos. Por ejemplo, si se crea un alias para la ruta /data, pero el archivo que se redirige se encuentra en /home/username, se producirá un error. Por lo tanto, es mejor crear alias que apunten a partes de su sistema de archivos al resto del mundo.

**NOTA:** Si utilizas un directorio que contiene una barra final, ten en cuenta que las direcciones URL que no tienen una barra final al final de la ruta no serán capaces de acceder al directorio sin él. Por lo tanto, si tu directorio tiene una barra final, es necesario que las direcciones URL que apuntan a él también la tengan.

## AliasMatch

La directiva AliasMatch es parecida a la directiva Alias, excepto en que puede utilizar expresiones regulares.

**Sintaxis:** AliasMatch regex ruta

**Contexto:** configuración del servidor, host virtual

Por ejemplo, la siguiente directiva hace corresponder www.yoursite.com/data/index.html con el archivo /web/data/index.html:

```
AliasMatch ^/data(.*) /web/data$1
```

## Redirect

La directiva Redirect redirige una solicitud URL a otra. Si ha movido una sección de su sitio Web a un nuevo directorio o, incluso, a un nuevo sitio Web, puede utilizar esta directiva para asegurarse de que la gente que tiene el antiguo sitio Web en la carpeta de favoritos seguirá siendo capaz de encontrarlo.

**Sintaxis:** Redirect [status\_code] URL-antigua URL-nueva

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorio (.htaccess)

Por ejemplo, la siguiente directiva redirige todas las URL solicitadas que contienen la ruta /data a una nueva URL. Por lo tanto, las solicitudes a www.yoursite.com/data/somefile.txt se redireccionarán a www.your-new-site.com/data/somefile.txt:

```
Redirect /data www.your-new-site.com/data
```

La directiva Redirect tiene preferencia sobre las directivas Alias y ScriptAlias. Por defecto, el código de estado enviado al cliente es Temp (código de estado HTTP 302). Si quiere especificar un código de estado distinto, utilice lo siguiente:

Código de estado	Lo que hace
Permanent	Le dice al cliente que el redirecciónamiento es permanente. Se devuelve el código de estado HTTP 301.

Código de estado	Lo que hace
Temp	Devuelve un estado temporal de redirecciónamiento (302). Este es el valor por defecto.
See other	Devuelve un estado See Other (303), indicando que ese recurso se ha reemplazado.
Gone	Devuelve un estado Gone (410) indicando que el recurso se ha eliminado de forma permanente. Cuando se utiliza este estado, se suele omitir el argumento de la URL.

**NOTA:** Puede proporcionar códigos de estado HTTP válidos en formato numérico. Si el estado que proporciona se encuentra entre 300 y 399, debe estar presente la nueva URL; en caso contrario, debe omitirse. Podría preguntarse sobre la utilización de los distintos códigos de estado. En el futuro, los sistemas cliente pueden ser lo suficientemente inteligentes como para reconocer los códigos de estado de una forma más significativa. Por ejemplo, si un servidor proxy recibe un código de estado con una redirección permanente, puede almacenar esta información en el caché de modo que sea capaz de acceder directamente al nuevo recurso en una solicitud posterior.

## RedirectMatch

La directiva RedirectMatch es parecida a la directiva Redirect, pero acepta expresiones regulares en lugar de una simple URL.

**Sintaxis:** RedirectMatch [código-estado] regex URL

**Contexto:** configuración del servidor, host virtual

Por ejemplo, la siguiente directiva redirige todas las solicitudes que terminan en .htm a una versión .html de la misma solicitud:

```
RedirectMatch (.*)\.htm$ www.yourserver.com$1.html
```

Como ejemplo del modo en el que esto podría funcionar, tenemos que la siguiente solicitud:

```
http://www.yoursite.com/some/old/dos/files/index.htm
```

es redirigida a:

```
http://www.yoursite.com/some/old/dos/files/index.html
```

Ver la directiva Redirect (última sección) para obtener información sobre status\_code.

## **RedirectTemp**

La directiva RedirectTemp es similar a la directiva Redirect. Permite al cliente saber que el redireccionamiento es temporal. Tenga en cuenta que la directiva Redirect también produce un estado temporal por defecto.

**Sintaxis:** RedirectTemp URL-antigua URL-nueva

**Contexto:** configuración del servidor, host virtual, directorio, control de acceso en el ámbito de directorio (.htaccess)

## **RedirectPermanent**

La directiva RedirectPermanent es parecida a la directiva Redirect. Permite que el cliente sepa que la redirección es permanente. Tenga en cuenta que la directiva Redirect produce un estado temporal por defecto, pero puede utilizar el código de estado 301 o la palabra clave permanent.

**Sintaxis:** RedirectPermanent URL-antigua URL-nueva

**Contexto:** configuración del servidor, host virtual, directorio, control de acceso en el ámbito de directorio (.htaccess)

## **ScriptAlias**

La directiva ScriptAlias crea un alias para la ruta física del directorio. Además, cualquier nombre de archivo suministrado en la solicitud es tratado como un script CGI, y el servidor intenta ejecutar el script.

**Sintaxis:** ScriptAlias alias "ruta-fisica-directorio"

**Contexto:** configuración del servidor, host virtual

Por ejemplo, la siguiente directiva puede utilizarse para procesar una solicitud del tipo `www.nitec.com/cgi-bin/somescript.pl`. El servidor intenta ejecutar `somescript.pl` si se verifica el permiso adecuado. Tenga en cuenta que el directorio ScriptAlias no es navegable:

```
ScriptAlias /cgi-bin/ "/www/nitec/public/cgi-bin/"
```

## **ScriptAliasMatch**

La directiva ScriptAliasMatch es equivalente a la directiva ScriptAlias excepto en que utiliza una expresión regular, que le permite definir una regla dinámica para alias, en lugar de un alias fijo.

**Sintaxis:** ScriptAliasMatch regex directorio

**Contexto:** Configuración del servidor, host virtual

Por ejemplo, las dos directivas siguientes hacen exactamente lo mismo:

```
ScriptAliasMatch ^/cgi-bin(.*) "/www/nitec/public/cgi-bin\$1"
ScriptAlias /cgi-bin/      "/www/nitec/public/cgi-bin/"
```

## mod\_speling

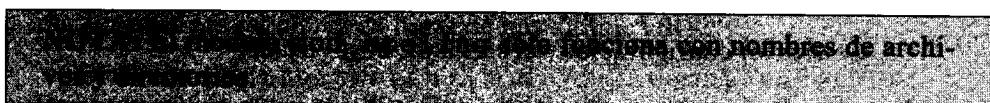
El módulo mod\_speling no está compilado en Apache por defecto. Le permite manejar las solicitudes URL mal escritas o con problemas de mayúsculas y minúsculas. Compara el nombre del documento solicitado con todos los nombres de documentos en el directorio solicitado que tengan una o más coincidencias.

En el caso de la solicitud de un documento mal escrito, el módulo permite un solo error, como por ejemplo, la inserción de un carácter extra, la omisión de un carácter o una transposición. En el caso de errores en las mayúsculas y minúsculas, realiza una comparación de archivos independiente de mayúsculas y minúsculas. En cualquiera de los dos casos, si el módulo localiza un solo documento que se parece realmente al solicitado, lo envía al cliente. Si hay más de una coincidencia, le manda al cliente una lista. La única directiva que ofrece este módulo es CheckSpelling. La directiva CheckSpelling activa o desactiva el módulo mod\_speling. Tenga en cuenta que cuando esta activada la corrección ortográfica, el servidor podría experimentar una pérdida de rendimiento debida a la realización de búsquedas extra que son necesarias para servir una solicitud de un documento mal escrito.

**Sintaxis:** CheckSpelling On | Off

**Predefinido:** CheckSpelling Off

**Contexto:** configuración del servidor, host virtual



## mod\_vhost\_alias

El módulo mod\_vhost\_alias le permite crear dinámicamente host virtuales configurados. Este módulo sólo es adecuado para instalaciones Apache que necesiten muchos host virtuales. Por ejemplo, un Internet Service Provider (ISP) utilizando Apache, puede utilizar este módulo para reducir el trabajo de configuración que, de otro modo, sería necesario para cada nuevo cliente del host virtual.

Este módulo le permite crear configuraciones dinámicas de host virtuales utilizando la dirección IP o el nombre del host de un sitio Web virtual en la creación de las rutas físicas de los directorios necesarias para servir el sitio.

### VirtualDocumentRoot

La directiva VirtualDocumentRoot le permite asignar la raíz de documentos para el host virtual utilizando una ruta de directorios interpolados.

**Sintaxis:** VirtualDocumentRoot directorio-interpolado

**Contexto:** configuración del servidor, host virtual

En la siguiente directiva, por ejemplo, cuando Apache recibe una solicitud para `http://www.domain.com/somepage.html`, la traduce a `/www/www.domain.com/htdocs/somepage.html`:

```
UseCanonicalName      Off  
VirtualDocumentRoot /www/%0/htdocs
```

`UseCanonicalName` está fijada en `off` por lo que Apache depende de la cabecera del Host para el nombre del host, que es suministrado por todos los clientes Web modernos.

La directiva `VirtualDocumentRoot` está indicada para los escenarios de alojamientos virtuales basados en el nombre, en los que tiene una dirección IP responsable de varios sitios Web.

**NOTA:** `%0` es traducido al nombre completo del host (es decir, `www.domain.com`). Si lo deseas, puedes utilizar partes del nombre del host. El nombre del host (o la dirección IP) se divide en partes separadas por puntos. Por ejemplo, utilice `%1` (primera parte = `www`), `%2` (segunda parte = `domain`), o `%-1` (última parte = `.com`) para crear los directorios interpolados apropiados para las directivas proporcionadas por este módulo. También puedes utilizar la convención `%N`, `P` en la que `N` representa una parte (separada por el punto) y `P` representa un número de caracteres de esa parte. Por ejemplo `%1.2` le dará `ww` de `www.domain.com`.

## **VirtualDocumentRootIP**

La directiva `VirtualDocumentRootIP` le permite asignar la raíz de documentos para el host virtual utilizando una ruta de directorios interpolados, que se construye utilizando la dirección IP del sitio Web. Este método es conveniente si utiliza un alojamiento virtual basado en IP, porque tiene una sola dirección IP para cada sitio Web virtual.

**Sintaxis:** VirtualDocumentRootIP directorio-interpolado

**Contexto:** configuración del servidor, host virtual

En la siguiente directiva, por ejemplo, cuando Apache recibe una solicitud para `http://www.domain.com/somepage.html`, traduce la solicitud a `/www/IP_address_of_www.domain.com/htdocs/somepage.html`:

```
VirtualDocumentRootIP /www/%0/htdocs
```

## **VirtualScriptAlias**

La directiva `VirtualScriptAlias` le permite definir un alias de script (al igual que la directiva `ScriptAlias`) que utiliza una ruta de directorio interpolado.

**Sintaxis:** `VirtualScriptAlias alias interpolated_directory`

**Contexto:** configuración del servidor, host virtual

En la siguiente directiva. Por ejemplo, cuando Apache recibe una solicitud para `http://www.domain.com/cgi-bin/script_name`, traduce la solicitud a `/www/www.domain.com/cgi-bin/script_name`:

```
UseCanonicalName      Off
VirtualScriptAlias   /cgi-bin/    /www/%0/cgi-bin
```

La directiva `UseCanonicalName` está fijada en `off`, y necesita que Apache dependa de la cabecera del Host para el nombre del host, cabecera que es suministrada por todos los clientes Web modernos.

El `VirtualDocumentRoot` es conveniente para los escenarios de alojamientos virtuales basados en nombre, en los que tiene una dirección de IP responsable de muchos sitios Web virtuales.

## **VirtualScriptAliasIP**

La directiva `VirtualScriptAliasIP` le permite definir un alias de script (al igual que hace la directiva `ScriptAlias`) que utiliza una ruta de directorio interpolada.

**Sintaxis:** `VirtualScriptAliasIP alias directorio-interpolado`

**Contexto:** Configuración del servidor, host virtual

En la siguiente directiva, por ejemplo, cuando se recibe en Apache una solicitud para `http://www.domain.com/cgi-bin/script_name`, Apache traduce la solicitud a `/www/IP_address/cgi-bin/script_nameVirtualScriptAliasIP /cgi-bin/ /www/%0/cgi-bin`.

## **Otros módulos**

Los módulos de esta sección, mostrados en la tabla 5.11, no se encuentran dentro de ninguna categoría en particular.

**Tabla 5.11.** Otros módulos

Módulo	Función
mod_so	Soporte para cargar módulos en tiempo de ejecución.
mod_imap	El manejador de archivos de integración de imágenes.
mod_proxy	Convierte a Apache en un servidor proxy con capacidad de caching. Ver el capítulo 10 para obtener los detalles.
mod_isapi	Soporte de la extensión ISAPI de Windows. Ver el capítulo 21 para obtener los detalles.
mod_file_cache	Archivos cacheados en la memoria para dar un servicio más rápido.
mod_dav	Proporciona funcionalidad Web-based Distributed Authoring and Versioning (WebDAV) de clase 1 y de clase 2.
mod_example	Un ejemplo de desarrollo de un módulo para aprender a escribir un módulo de Apache. Sólo es útil para los programadores de C.

## **mod\_so**

El módulo mod\_so permite a Apache cargar el código ejecutable que necesitan otros módulos o cargar otros módulos durante la puesta en marcha del servidor. Puede compilar todos los módulos como módulos DSO (Dynamic Shared Object) excepto éste.

### **LoadFile**

La directiva LoadFile carga el archivo nombrado durante el arranque. Normalmente, se carga un archivo DLL (dynamically linked library) que necesita otro módulo utilizando esta directiva (sólo en Windows).

**Sintaxis:** LoadFile nombrearchivo [nombrearchivo...]

**Contexto:** configuración del servidor

### **LoadModule**

La directiva LoadModule carga un módulo que se ha compilado como un DSO.

**Sintaxis:** LoadModule module\_filename

**Contexto:** configuración del servidor

## mod\_imap

El módulo mod\_imap se compila en Apache por defecto. Proporciona soporte de integración de imágenes, que ha sido proporcionada por el programa CGI imagemap. Puede utilizar la directiva AddHandler para especificar el manejador de imap-file (construido en este módulo) para cualquier extensión de archivo.

Por ejemplo, la siguiente directiva hace que Apache trate a todos los archivos que tengan la extensión .map como integración de imágenes, y los Apache procesa los archivos utilizando el módulo mod\_imap:

```
AddHandler imap-file map
```

**NOTA:** El módulo mod-imap sigue soportando el antiguo formato:

```
AddType application/x-httd-imap.map
```

**Sin embargo, no se recomienda el formato antiguo porque su soporte posiblemente se rechazará en el futuro.**

Los contenidos de un archivo de integración de imágenes pueden tener cualquiera de las siguientes sintaxis:

```
directive value [x,y ...]  
directive value "Menu text" [x,y ...]  
directive value x,y ... "Menu text"
```

Las directivas permitidas en un archivo de integración de imágenes son:

- **base:** las URL relativas que se utilizan en los archivos de integración se consideran relativas al valor de esta directiva. Tenga en cuenta que la asignación de la directiva Imapbase se invalida por esta directiva cuando se encuentra en un archivo de integración. Su valor por defecto es `http://server_name/. base_uri`, que es sinónimo de base.
- **default:** especifica la acción a tomar cuando las coordenadas no se corresponden con las de poly, circle o rect, y no se dan directivas point. El valor por defecto para esta directiva es nocontent, que le dice al cliente que mantenga la misma página desplegada.
- **poly:** define un polígono utilizando al menos 3 puntos hasta un máximo de 100 puntos. Si el usuario que proporciona las coordenadas cae dentro del polígono, se activa esta directiva.

- **circle**: define un círculo utilizando el centro de coordenadas y un punto del círculo. Si el usuario que proporciona las coordenadas cae dentro del círculo, se activa esta directiva.
- **rect**: define un rectángulo utilizando las coordenadas de dos esquinas opuestas. Si el usuario que proporciona las coordenadas cae dentro del rectángulo, se activa esta directiva.
- **point**: define una sola coordenada puntual. La directiva **point** más cercana a la coordenada suministrada por el usuario, se utiliza cuando no se satisface ninguna otra directiva.

El valor es una URL absoluta o relativa, o uno de los valores especiales en la lista siguiente. Las coordenadas (x,y) están separadas por caracteres en blanco. El texto que va entre comillas dobles (mostrado en el segundo tipo de sintaxis) se utiliza como texto del enlace, si se genera un menú de integración de imágenes. Cualquier línea con un carácter almohadilla # se considera como un comentario y Apache lo ignora. Las coordenadas están escritas en el formato x,y, en el que cada coordenada está separada por un carácter en blanco. La cadena de texto entre comillas se utiliza como enlace cuando se genera un menú. En ausencia de dicha cadena, la URL es el enlace, tal y como se muestra en el siguiente archivo de integración de imágenes:

```
# Los comentarios van aquí
# Versión 1.0.0
base http://www.yoursite.com/some/dir
rect thisfile.html "Customer info" 0,0 100,200
circle http://download.yoursite.com/index.html 295,0 100,22
```

Si este archivo de integración de imágenes se llama `imagemap.map`, puede venir referido como se muestra a continuación desde otro archivo HTML:

```
<A HREF="/path/to/imagemap.map"><IMG ISMAP SRC="/path/to/
imagemap.gif"></A>
```

## **ImapMenu**

La directiva `ImapMenu` determina la acción para una solicitud de un archivo de integración de imágenes sin coordenadas válidas.

**Sintaxis:** `ImapMenu {None, Formatted, Semi-formatted, Unformatted}`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorio (`.htaccess`)

**Invalidar:** `Indexes`

`ImapMenu` permite las siguientes acciones:

Acción	Lo que hace
None	No se genera ningún menú, y se lleva a cabo la acción por defecto.
Formatted	Se genera el menú más sencillo. Se ignoran los comentarios. Se imprime una cabecera de nivel 1, y después una regla horizontal y los enlaces, en líneas separadas.
Semi-formatted	En el menú semi-formateado, se imprimen los comentarios, las líneas en blanco se convierten en saltos HTML y no se imprime ni una cabecera ni una regla horizontal.
Unformatted	En el menú sin formato, se imprimen los comentarios y se ignoran las líneas en blanco.

## ImapDefault

La directiva `ImapDefault` define la acción por defecto para la integración de imágenes. Este valor por defecto se puede invalidar en el archivo de integración de imágenes utilizando la directiva por defecto.

**Sintaxis:** `ImapDefault {Error, Nocontent, Map, Referer, URL}`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorio (`.htaccess`)

**Invalidar:** `Indexes`

La siguiente tabla muestra el significado de cada uno de los posibles valores para la directiva `ImapDefault`.

Valor	Lo que significa
URL	Una URL relativa o absoluta. Las URL relativas resuelven en relación con la base.
Map	Lo mismo que URL del archivo de integración de imágenes en sí. A no ser que <code>ImapMenu</code> tenga un valor asignado de <code>none</code> , se creará un menú.
Menu	Lo mismo que Map.
Referer	Lo mismo que URL del documento al que se refiere. El valor por defecto es <code>http://servername/</code> si no está presente <code>Referer: header</code> .

Valor	Lo que significa
Nocontent	Se envía un código de estado de 204 para decirle al cliente que mantenga la misma página desplegada. No es válido para base.
Error	Se envía un código de estado de 500 para informar al cliente sobre un error del servidor.

## ImapBase

La directiva ImapBase asigna la base por defecto que se utiliza en los archivos de integración de imágenes. Esta base asignada se puede invalidar utilizando la directiva base dentro del archivo de integración de imágenes. Si esta directiva no está presente, la base por defecto es `http://servername/`.

**Sintaxis:** ImapBase {Map, Referer, URL}

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorio (`.htaccess`)

**Invalidar:** Indexes

## mod\_file\_cache

El módulo mod\_file\_cache permite a Apache guardar archivos estáticos frecuentemente utilizados en el caché que no se cambian a menudo. Utilizando este módulo, puede cargar un archivo en la memoria o preabrir un archivo y reducir la cantidad de entradas y salidas en el disco para cada solicitud. Esto significa que Apache no tiene que leer archivos desde el disco para cada solicitud. Este módulo no funciona en todas las plataformas y debería utilizarse únicamente si está seguro de que ciertos archivos solicitados con frecuencia no se van a cambiar. Si un archivo cacheado o un archivo preabierto cambia, el servidor Apache no puede admitir los cambios hasta que no sea reiniciado.

## MMapFile

La directiva MMapFile le permite cargar un archivo en la memoria mediante llamadas sistemáticas `mmap()`. Una vez que se ha cargado un archivo en la memoria, el servidor no detectará ningún cambio en el archivo físico. Por lo tanto, es necesario tener cuidado cuando se utiliza esta directiva. Si cambia un archivo que se ha cargado previamente en la memoria, asegúrese de reiniciar el servidor Apache.

**Sintaxis:** MmapFile nombrearchivo [nombrearchivo] [...]

**Contexto:** configuración del servidor

## **CacheFile**

La directiva CacheFile preabre un archivo y lo guarda en el caché, de modo que cuando llega la solicitud, Apache no tiene que realizar llamadas sistemáticas para abrir y leer el archivo. Una vez que se ha guardado un archivo en el caché, cualquier cambio que se quiera realizar en el archivo requerirá reiniciar el servidor Apache.

**Sintaxis:** CacheFile nombrearchivo [nombrearchivo] [...]

**Contexto:** configuración del servidor

## **mod\_dav**

El módulo mod\_dav le permite utilizar las extensiones WebDAV del protocolo HTTP 1.1. Para aprender más sobre estas extensiones visite el sitio [www.webdav.org](http://www.webdav.org).

### **Dav**

La directiva Dav valida o invalida el módulo mod\_dav. Debe asignarse el valor On si quiere utilizar la característica WebDAV dentro de un contenedor de directivas.

**Sintaxis:** Dav On | Off

**Contexto:** directorio

### **DavLockDB**

La directiva DavLockDB asigna el fully qualified pathname (nombre completo de la ruta de la máquina, incluido el dominio) del archivo de bloqueo de bases de datos.

**Sintaxis:** DavLockDB nombrearchivo

**Contexto:** configuración del servidor, host virtual

### **DavMinTimeout**

La directiva DavMinTimeout fija el intervalo de bloqueo de recursos mínimo en segundos. El valor por defecto asegura que un cliente WebDAV no es bloqueado (independientemente del tiempo) automáticamente por el servidor.

**Sintaxis:** DavMinTimeout seconds

**Predefinido:** DavMinTimeout 0

**Contexto:** directorio

## **DavDepthInfinity**

La directiva DavDepthInfinity permite que una solicitud PROPFIND con la cabecera Depth se fije en infinito. Se recomienda el valor por defecto.

**Sintaxis:** DavDepthInfinity On | Off

**Predefinido:** DavDepthInfinity Off

**Contexto:** directorio



# **Parte II**

# **Administrar**

# **sitios Web**



# 6 Alojar sitios Web virtuales

---

## En este capítulo

1. Conocemos los dos tipos de host virtuales.
2. Creamos host virtuales basados en IP.
3. Creamos host virtuales basados en nombre.
4. Configuramos DNS para host virtuales.
5. Ejecutamos un sitio Web con sus propios privilegios de usuario y de grupo.
6. Manejamos una gran cantidad de host virtuales en un solo servidor Apache utilizando `mod_perl`.
7. Simplificamos el proceso de creación de nuevos host virtuales utilizando el script `makesite`.
8. Manejamos host virtuales utilizando el módulo `mod_v2h` y la base de datos MySQL.

Apache puede atender varios sitios Web desde un único servidor. Por ejemplo, una compañía Web alojada puede tener un solo servidor Web ejecutando

Apache, que está sirviendo a cientos de sitios Web clientes. Este tipo de sistema tiene un nombre de host principal y muchos alias IP o nombres de host virtuales. Un sitio Web atendido mediante este tipo de host virtuales se denomina sitio Web virtual. El soporte de Apache para sitios virtuales (llamados hosts virtuales) es impresionante. Este capítulo discute el modo de crear varios tipos de host virtuales y cómo manejarlos utilizando distintas técnicas.

## Entender las capacidades del hospedaje virtual en Apache

Cuando establece Apache en un host de Internet, puede responder a una solicitud HTTP a ese host. Por ejemplo, si establece Apache en un host llamado `server1.doman.com`, Apache servirá solicitudes HTTP a ese host. Sin embargo, si establece sus registros DNS con dos nombres de host (como por ejemplo, `www.mycompany-domain.com` y `www.friendscompany-domain.com`) en la misma máquina, Apache puede servir a estos dos dominios como sitios Web virtuales. En ese caso, `www.mycompany-domain.com` se considera el nombre del host Web principal (servidor principal), y el resto se considerarán como sitios Web virtuales o host virtuales.

Apache permite que los host virtuales hereden configuración del servidor principal, lo que da lugar a una configuración del host virtual mucho más manejable en grandes instalaciones, en las que se puede compartir parte del contenido. Por ejemplo, si decide tener únicamente un depósito central de CGI y permitir a los host virtuales utilizar los script que están almacenados en ese depósito, no necesita crear una directiva `ScriptAlias` en cada contenedor de host virtual. Simplemente utilice una directiva en la configuración del servidor principal y tendrá todo el trabajo hecho. Cada host virtual puede utilizar el alias como si le perteneciese.

El archivo de configuración de Apache, el `httpd.conf`, separa la configuración del host virtual de la configuración del servidor principal utilizando el contenedor `<VirtualHost>`. Por ejemplo, observe el archivo `httpd.conf` del listado 6.1.

### Listado 6.1. `httpd.conf`

```
# archivo httpd.conf

ServerName main.server.com
Port 80
ServerAdmin mainguy@server.com
DocumentRoot "/www/main/htdocs"

ScriptAlias /cgi-bin/ "/www/main/cgi-bin/"
```

```

Alias /images/ "/www/main/htdocs/images/"

<VirtualHost 192.168.1.100>
    ServerName vhost1.server.com
    ServerAdmin vhost1_guy@vhost1.server.com
    DocumentRoot "/www/vhost1/htdocs"
    ScriptAlias /cgi-bin/ "/www/vhost1/cgi-bin/"
</VirtualHost>

<VirtualHost 192.168.1.110>
    ServerName vhost2.server.com
    ServerAdmin vhost2_guy@vhost2.server.com
    DocumentRoot "/www/vhost2/htdocs"
    ScriptAlias /cgi-bin/ "/www/vhost2/cgi-bin/"
    Alias /images/ "/www/vhost2/htdocs/images/"
</VirtualHost>
```

El listado 6.1 muestra dos sitios Web virtuales llamados `vhost1.server.com` y `vhost2.server.com`, que están definidos en sus propios contenedores `<VirtualHost>`. Todas las directivas incluidas en cada uno de los contenidores `<VirtualHost>` se aplican, únicamente, al host virtual al que sirven. Por tanto, cuando un navegador Web solicita `http://vhost1.server.com/index.html`, el servidor Web Apache busca la página `index.html` en el directorio `/www/vhost1/htdocs`. De igual modo, cuando un navegador Web solicita `http://vhost2.server.com/cgi-bin/hello.pl`, el script se ejecuta desde el directorio `/www/vhost2/cgi-bin`.

Sin embargo, muchas directivas de la configuración del servidor principal (es decir, cualquier directiva fuera del contenedor `<VirtualHost>`) se siguen aplicando al host virtual que no las invalida. Por ejemplo, el servidor `vhost1.server.com` del listado 6.1 no tiene una directiva `Alias` para el alias del directorio `/images/`. Por lo tanto, cuando el navegador Web solicita el archivo `http://vhost1.server.com/images/pretty_pic.gif`, la imagen se sirve desde el directorio `/www/main/htdocs/images`. Como `vhost2.server.com` invalida el alias `/images/` por su cuenta, se sirve una solicitud similar desde el directorio `/www/vhost2/htdocs/images`.

## Establecer un host virtual

Hay tres modos de crear sitios Web virtuales cuando utilizamos Apache:

- **Basado en el nombre:** los sitios Web virtuales basados en el nombre son muy comunes. Este tipo de configuración exige que tenga varios nombres de host en un solo sistema. Puede crear varios CNAME o registros A en DNS para que se encuentren en un solo host. Como este método no utiliza direcciones IP en la configuración de Apache, es fácil de portar si cambia sus direcciones IP a su servidor Web.

- **Basado en IP:** este método necesita direcciones IP en la configuración de Apache y por eso resulta sencillo de portar cuando hay que cambiar las direcciones IP.
- **Varios servidores principales:** este método implica la utilización de varias configuraciones de servidores Web principales. Este método sólo se recomienda en el caso de que deba mantener archivos de configuración separados para host virtuales. Este es el método menos recomendado y es difícil de usar.

## Host virtuales basados en nombre

Este es el método más recomendado. Necesita una sola dirección IP para alojar cientos de sitios Web virtuales.

**NOTA:** Lo único que ha de recordar cuando utilice este método, es que no funciona con los navegadores Web que no soportan el protocolo HTTP 1.1. Únicamente los primeros navegadores, como Microsoft IE 1.x o Netscape Navigator 1.x, no soportan HTTP 1.1. Por lo que, realmente, no se trata de un problema muy grave. La mayoría de la gente utiliza 3.x o versiones posteriores de navegadores Web, que son compatibles con la técnica de hospedaje virtual basada en nombre.

Por ejemplo, imagine que tiene una dirección IP 192.168.1.100 y quiere alojar vhost1.domain.com y vhost2.domain.com en el mismo servidor. Puede hacer lo siguiente:

1. Primero, tiene que crear el registro DNS apropiado en su servidor DNS para enlazar vhost1.domain.com y vhost2.domain.com a la dirección IP 192.168.1.100. Diríjase a la sección "Configurar DNS para un host virtual" para obtener los detalles.
2. Tiene que crear un segmento de configuración, parecido al que le presentamos a continuación, en el archivo httpd.conf.

```
NameVirtualHost 192.168.1.100

<VirtualHost 192.168.1.100>
    ServerName vhost1.domain.com
    ServerAdmin someone@vhost1.domain.com
    DocumentRoot "/www/vhost1/htdocs"

    #
    # Aquí se coloca cualquier directiva extra que necesite
    #
```

```

    </VirtualHost>

<VirtualHost 192.168.1.100>
    ServerName vhost2.domain.com
    ServerAdmin someone@vhost2.domain.com
    DocumentRoot "/www/vhost2/htdocs"

    #
    # Aquí se coloca cualquier directiva extra que necesite
    #

</VirtualHost>

```

No olvide crear el directorio raíz de documentos si aún no lo ha hecho. Además, puede añadir más directivas en cada una de las configuraciones de los host virtuales en caso necesario.

3. Reinicie Apache utilizando el comando de reinicio /usr/local/apache/apachectl y acceda a cada uno de los host virtuales utilizando <http://vhost1.domain.com> y <http://vhost2.domain.com>.

Como acabamos de ver en el ejemplo, ambos contenedores de host utilizan la misma dirección IP (192.168.1.100). Por lo tanto, debemos preguntarnos cómo sabe Apache cuál es el sitio Web que se está solicitando cuando llega una solicitud por la dirección IP 192.168.1.100.

HTTP 1.1 necesita que una cabecera llamada Host esté presente en cada solicitud que el navegador le presenta al sitio Web. Por ejemplo, a continuación tenemos una cabecera de una solicitud HTTP de un navegador Web a un servidor ejecutándose en rhat.domain.com.

```

GET / HTTP/1.1
Host: rhat.domain.com
Accept: text/html, text/plain
Accept: postscript-file, default, text/sgml, */*;q=0.01
Accept-Encoding: gzip, compress
Accept-Language: en
User-Agent: Lynx/3.0.0dev.9 libwww-FM/2.14

```

Cuando Apache ve la cabecera Host: rhat.domain.com, puede servir de forma inmediata, la solicitud utilizando el host virtual apropiado, es decir aquel que tiene un ServerName coincidente.

## Host virtuales basados en IP

Este método exige el uso de direcciones IP en la creación de los host. La dirección IP ha de estar codificada por hardware en el archivo de configuración en cada etiqueta del contenedor <VirtualHost>. Esto puede crear grandes quebraderos de cabeza si cambia las direcciones IP con cierta frecuencia. Este

método no tiene ninguna ventaja respecto al método anterior. El siguiente ejemplo muestra tres host virtuales basados en IP.

```
<VirtualHost 192.168.1.1>
    ServerName vhost1.server.com
    # Aquí se colocan otras directivas
</VirtualHost>

<VirtualHost 192.168.1.2>
    ServerName vhost2.server.com
    # Aquí se colocan otras directivas
</VirtualHost>

<VirtualHost 192.168.1.3>
    ServerName vhost3.server.com
    # Aquí se colocan otras directivas
</VirtualHost>
```

Cada una de estas direcciones IP, deben ir unidas a la interfaz Ethernet apropiada en el servidor. Por ejemplo, la configuración anterior requiere un sistema que aloje los sitios Web para tener los siguientes registros DNS en su archivo de configuración del servidor DNS.

```
; Address Records
vhost1.server.com.      IN      A 192.168.1.1
vhost2.server.com.      IN      A 192.168.1.2
vhost3.server.com.      IN      A 192.168.1.3

; Reverse DNS records
1                      IN PTR  vhost1.server.com.
2                      IN PTR  vhost2.server.com.
3                      IN PTR  vhost3.server.com.
```

Cada una de estas direcciones deben estar unidas a una o más interfaces Ethernet en el servidor. En un sistema Linux, se pueden unir varias direcciones IP utilizando la técnica de denominación IP (aliasing IP). Por ejemplo:

```
/sbin/ifconfig eth0 192.168.1.1 up
/sbin/ifconfig eth0:0 192.168.1.2 up
/sbin/ifconfig eth0:1 192.168.1.3 up
```

Las tres direcciones IP están unidas a la interfaz Ethernet `eth0` y a sus dos alias, `eth0:0` y `eth0:1`. Como consecuencia, el sistema responderá a cada dirección IP.

## Varios servidores principales como host virtuales

Sólo se recomienda utilizar varios servidores principales como host virtuales cuando está obligado (normalmente, por razones que no son de índole técnico) a

mantener distintos archivos de configuración `httpd.conf`. Por ejemplo, imagine que tiene 16 direcciones IP y que quiere suministrar a 16 clientes (o departamentos) su propio archivo de configuración `httpd.conf` para, que de ese modo, cada entidad pueda gestionar absolutamente todo por su cuenta e, incluso, crear un host virtual utilizando los contenedores `<VirtualHost>` dentro de su propio `httpd.conf`.

**NOTA:** Antes de adentrarse en la aventura de utilizar este método, considere con cuidado, la posibilidad de evitar crear varias instancias de servidores principales utilizando los contenedores `<virtualhost>`.

El listado 6.2 muestra una versión simplificada de un `httpd.conf` (llamado `httpd-100.conf`) que utiliza la directiva `Listen` para decirle a Apache que únicamente sirva la dirección IP `192.168.1.100`, asociada con el sistema que está ejecutándose. Esto da lugar a la implementación de un solo host virtual.

#### Listado 6.2. `httpd-100.conf`

```
ServerType standalone
ServerRoot "/usr/local/apache"
PidFile /usr/local/apache/logs/httpd-192.168.1.100.pid
ScoreBoardFile /usr/local/apache/logs/httpd-
192.168.1.100.scoreboard
Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15
MinSpareServers 60
MaxSpareServers 100
StartServers 50
MaxClients 200
MaxRequestsPerChild 0
Port 80
Listen 192.168.1.100:80
User prod
Group prod
ServerName prod.domain.com
ServerAdmin kabir@prod.domain.com
DocumentRoot "/www/prod/htdocs"
```

Tenga en cuenta que la directiva `Listen` también toma un número de puerto como parámetro. En el listado 6.2, se le dice a Apache que escuche la dirección IP dada en el puerto 80. La directiva `Port` sigue siendo necesaria porque su valor se utiliza en las URL de referencia propia generadas.

El listado 6.3 muestra otro archivo `httpd.conf` simplificado (llamado `httpd-101.conf`) que le dice a Apache que escuche únicamente la dirección `192.168.1.10`. Esto da lugar a la implementación de otro host virtual

utilizando el método de varios servidores principales en la creación de host virtuales.

#### Listado 6.3. httpd-101.conf

```
ServerType standalone
ServerRoot "/usr/local/apache"
PidFile /usr/local/apache/logs/httpd-192.168.1.101.pid
ScoreBoardFile /usr/local/apache/logs/httpd-
192.168.1.101.scoreboard
Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15
MinSpareServers 5
MaxSpareServers 10
StartServers 5
MaxClients 10
MaxRequestsPerChild 0
Port 80
Listen 192.168.1.101:80
User stage
Group stage
ServerAdmin lance@stage.domain.com
DocumentRoot "/www/stage/htdocs"
```

Un sistema que quiera ejecutar dos servidores principales Apache, utilizando los archivos de configuración de los listados 6.2 y 6.3, debe tener:

- Una o más interfaces Ethernet respondiendo a las direcciones IP nombradas. Por ejemplo, en un sistema Lynux, puede unir la interfaz Ethernet eth0 tanto a 192.168.1.100 como a 192.168.1.101 utilizando el comando denominación (aliasing) IP, del siguiente modo:

```
/sbin/ifconfig eth0 192.168.1.100 up
/sbin/ifconfig eth0:0 192.168.1.101 up
```

Por supuesto, si un sistema tiene varias interfaces Ethernet y quiere utilizar un servidor principal para cada interfaz, no necesita utilizar alias IP.

- Las direcciones IP deben tener nombres de host asociados con ellas. Para los archivos de configuración anteriores, la dirección 192.168.1.100 está asociada con prod.domain.com y la dirección 192.168.1.101 está asociada a stage.domain.com.

Un sistema que tenga esas direcciones IP unidas a su interfaz o interfaces, puede ejecutar dos demonios principales de Apache utilizando los siguientes comandos:

```
/usr/local/apache/bin/httpd -f conf/httpd-100.conf
/usr/local/apache/bin/httpd -f conf/httpd-101.conf
```

Si está ejecutando el comando `ps auxww | grep httpd | grep root | grep conf`, verá dos servidores principales Apache que se ejecutan como usuario raíz.

## Configurar DNS para un host virtual

En la mayor parte de los casos, su ISP es responsable de proporcionar servicio DNS para dominios. En este caso, puede saltarse esta sección. Sin embargo, si no ejecuta BIND, el servidor DNS más utilizado, en Linux o en otro sistema del tipo Linux, le mostrará cómo configurar DNS para sus host virtuales.

A lo largo de este libro, los usuarios de Windows: pueden obtener los detalles de la configuración en su servidor DNS de Windows.

### Entender los archivos de zona

Es necesario un *archivo de zona* para establecer DNS para un host virtual. Un archivo de zona es una descripción textual de sus registros DNS. El servidor DNS carga estos archivos para implementar su servicio DNS para una sola zona, que normalmente denominamos dominio Internet.

A continuación tenemos un ejemplo de un archivo de zona. En este ejemplo, supongo que el nuevo host virtual que quiere establecer se llama `www.newdomain.com`, que los nombres de host del servidor son `ns1.domain.com` (principal) y `ns2.domain.com` (secundario), y que el nombre del host en su servidor Web es `www.domain.com` (192.168.1.100). Asegúrese de que cambia los nombres de host y las direcciones IP para tener el mismo establecimiento.

El archivo de zona para `www.newdomain.com` se llama `/var/named/newdomain.zone` y contiene las líneas siguientes:

```
@ IN SOA newdomain.com. hostmaster.newdomain.com. (
    20011201001 ; Serial YYYYMMDDXXX
    7200          ; refresh
    3600          ; (1 hour) retry
    1728000       ; (20 days) expire
    3600)         ; (1 hr) minimum ttl

; Nombres de los servidores
IN NS      ns1.domain.com.
IN NS      ns2.domain.com.

; Registros A para alojamiento virtual basado en IP
; www.newdomain.com.      IN   A     192.168.1.100

; CNAME para alojamiento virtual basado en nombre
www.newdomain.com.      IN   CNAME  www.domain.com.
```

**NOTA:** La configuración DNS anterior, supone que no va a utilizar el alojamiento virtual basado en IP y, por lo tanto, no crea un registro A para `www.newdomain.com`. La línea de registro A se comenta aparte. Puede borrar la señal de comentario si utiliza el método de alojamiento virtual basado en IP. Como el alojamiento virtual basado en nombre no necesita una dirección IP única, se crea un registro CNAME para relacionar `www.newdomain.com` con `www.domain.com` (que es el servidor Web principal).

A continuación tenemos lo que está ocurriendo en la configuración anterior:

- La primera línea comienza con un registro DNS llamado Start of Address (SOA), que especifica el número de serie, la razón de actualización, la razón de reintento, el tiempo de expiración y el tiempo de vida (TTL) de los valores.
- Los servidores DNS utilizan el número de serie para determinar si necesitan o no sus registros cacheados. Para entenderlo, en el ejemplo tenemos el número de serie 20011201001, que indica que la última actualización tuvo lugar el 12/01/2001 y que su primer (001) cambio tuvo lugar ese día. Si el administrador de DNS cambia la configuración DNS el día 12/02/2001, el número de serie debería cambiar para reflejar el cambio, de modo que cualquier servidor DNS remoto que tenga cacheados los registros, puede comparar números de serie entre la versión cacheada y la nueva versión, y decidir si descarga los datos DNS nuevos.
- La razón de actualización nos dice la frecuencia con que deberían actualizarse los registros en el servidor.
- La razón de reintento establece que en caso de un fallo en el proceso de pedida de solicitudes por parte de un servidor DNS remoto, este servidor DNS reintenta enviar la solicitud cada un cierto intervalo.
- El tiempo de expiración, le dice al servidor DNS remoto que elimine, a la fuerza, cualquier dato del caché que tenga en el dominio dado una vez que pasan una cantidad de días determinados.
- La entrada final en SOA, establece que los registros tengan un valor mínimo especificado de tiempo de vida. Tenga en cuenta que todo lo que se encuentre detrás del punto y coma es tratado como una línea de comentario y, por lo tanto, ignorado.
- Las siguientes dos líneas no comentadas establecen que los servidores DNS responsables de `newdomain.com` son `ns1.domain.com` y `ns2.domain.com`. Si no tiene un segundo servidor DNS, debería considerar el uso de un servicio DNS secundario de terceras partes como es el `http://www.secondary.com`.

## **Establecer las DNS para host virtuales nuevos**

Puede establecer un DNS para cada nuevo host virtual en Linux del siguiente modo:

1. Cree un archivo de zona para el nuevo dominio. (Ver la sección anterior para obtener los detalles sobre los archivos de zona.)
2. Añada las líneas que se encuentran a continuación, en el archivo /etc/named.conf, para permitir la zona newdomain.

```
zone "newdomain.com" IN {  
    type master;  
    file "newdomain.zone";  
    allow-update { none; };  
};
```

Esto le dice al demonio DNS que el archivo de zona de newdomain.com es /var/named/newdomain.zone, y que se trata de su servidor DNS principal (maestro), para esta zona.

3. Ejecute el comando killall -HUP named para forzar al servidor a recargar el /etc/named.conf y el nuevo archivo de zona.
4. Intente hacer un ping www.newdomain.com. Si no obtiene una respuesta, compruebe el archivo /var/log/messages para encontrar los errores que pueda tener el servidor (named\_) registrados en los archivos de zona /etc/named.conf o /var/named/newdomain. En ese caso, corrija los errores y reinicie el servidor tal y como se ha discutido en el paso previo.
5. Una vez que puede realizar un ping a www.newdomain.com, está preparado para crear configuración Apache para este host virtual, del modo descrito en secciones anteriores de este capítulo.

## **Ofrecer servicios de correo virtual**

Para proporcionar servicios de correo virtual en los host virtuales nuevos, tiene que modificar el archivo newdomain.zone y añadir uno o más registros MX apropiados. Por ejemplo, imagine que su servidor de correo es mail.domain.com y que está configurado para aceptar correos de newdomain.com. En este caso, puede modificar /var/named/newdomain.zone para que sea del siguiente modo:

```
@ IN SOA newdomain.com. hostmaster.newdomain.com. (   
    20011201002 ; Serial YYYYMMDDXXX  
    7200          ; refresh  
    3600          ; (1 hour) retry  
    1728000       ; (20 days) expire  
    3600)         ; (1 hr) minimum ttl
```

```
; Nombres de servidores
IN NS      ns1.domain.com.
IN NS      ns2.domain.com.
IN MX    10 mail.domain.com.

; CNAME para el alojamiento virtual basado en nombre
www.newdomain.com.      IN      CNAME      www.domain.com.
```

Tenga en cuenta que si tiene varios servidores de correo, puede añadirlos utilizando un esquema sencillo de prioridades. Por ejemplo, imagine que quiere que todos los correos de newdomain.com vayan a mail.domain.com, y que en caso de que este servidor esté incapacitado, quiere utilizar bkupmail.domain.com.

Puede añadir simplemente un segundo registro MX, tal y como se muestra a continuación:

```
IN MX    10 mail.domain.com.
IN MX    20 bkupmail.domain.com.
```

Al ser más pequeño el número asignado a mail.domain.com, le convierte en un servidor Web de mayor prioridad que el servidor bkupmail.domain.com. Asegúrese de que tiene configurado el software de su servidor de correo para que soporte dominios de correo virtuales.

## Asignar usuario y grupo a cada host virtual

Si ha configurado Apache utilizando el módulo MPM Perchild, puede asignar usuario y grupo para cada host virtual. El principal beneficio de este método es su simpleza y no el sistema de múltiples servidores de correo basados en httpd.conf.

**ADVERTENCIA:** Esta aproximación es más lenta que la ejecución de varios servidores principales Apache con distintas direcciones IP y distintas directivas User y Group. La pérdida de velocidad se debe al incremento de complejidad en el procesamiento de solicitudes internas. El concepto de este método es el siguiente: instruye a Apache para que ejecute un conjunto de procesos hijo con un ID de usuario y un ID de grupos determinados. Cuando llega una solicitud al host virtual y la toma un proceso hijo, éste determina primero si puede manejarla. Si el hijo no es responsable del host virtual solicitado deberá pasar la solicitud al hijo apropiado mediante una llamada socket, que disminuirá la velocidad del proceso de solicitud. Como este es un concepto nuevo, en futuras versiones Apache se solucionará el tema de velocidades.

El material que se presenta a continuación supone que quiere establecer dos host virtuales llamados `vhost1.domain.com` y `vhost2.domain.com`. Asegúrese de reemplazarlos con los nombres que utilice. La tabla 6.1 muestra los ID de usuario y de grupo que tiene que se utilizan para estos dominios virtuales.

**Tabla 6.1.** Las ID de usuario y de grupo en los host virtuales

Host virtual	ID de usuario	ID de grupo
<code>vhost1.domain.com</code>	<code>vh1user</code>	<code>vh1group</code>
<code>vhost2.domain.com</code>	<code>vh2user</code>	<code>vh2group</code>

Se puede configurar Apache para que soporte distintos ID de usuario y de grupo para cada host virtual del siguiente modo:

1. Añada las líneas siguientes a `httpd.conf`:

```
ChildPerUserID 10 vh1user vh1group  
ChildPerUserID 10 vh2user vh2group
```

La directiva `ChildPerUser` le dice a Apache que asocie diez procesos hijo (cada uno con varios hilos que sirven las solicitudes) al ID de usuario y de grupo del sitio `vhost1.domain.com`. Del mismo modo, la segunda línea asocia otros diez procesos hijo al ID de usuario y de grupo del sitio `vhost2.domain.com`.

2. Cree un contenedor `<VirtualHost>` para cada sitio:

```
NameVirtualHost 192.168.1.100  
  
<VirtualHost 192.168.1.100>  
  ServerName vhost1.domain.com  
  AssignUserID vh1user vh1group  
  #  
  # Aquí van otras directivas  
  #  
  </VirtualHost>  
  
  <VirtualHost 192.168.1.100>  
  ServerName vhost2.domain.com  
  AssignUserID vh2user vh2group  
  
  #  
  # Aquí van otras directivas  
  #  
  </VirtualHost>
```

La directiva `AssignUserID` utilizada en cada configuración del host virtual, le dice a Apache que asocie cada uno de los host virtuales al ID de

usuario y de grupo apropiados. No olvide cambiar la dirección IP en caso necesario.

3. Reinicie el servidor Apache utilizando el comando de reinicio /usr/local/apache/bin/apachectl.

## Gestionar un gran número de host virtuales

Si tiene muchos host virtuales, el archivo httpd.conf puede convertirse en un archivo muy grande y difícil de manejar. Una solución sencilla es poner cada host virtual en un archivo e incluir cada archivo utilizando la directiva `Include`. Por ejemplo, el listado 6.4 muestra un archivo httpd.conf con dos configuraciones de host virtual que son externas al archivo de configuración principal.

**Listado 6.4.** httpd.conf con dos configuraciones externas de host virtuales

```
ServerType standalone
ServerRoot "/usr/local/apache"
PidFile /usr/local/apache/logs/httpd.pid
ScoreBoardFile /usr/local/apache/logs/httpd.scoreboard
Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15
MinSpareServers 5
MaxSpareServers 10
StartServers 5
MaxClients 10
MaxRequestsPerChild 0
Port 80
User httpd
Group httpd
ServerName www.domain.com
ServerAdmin webmaster@domain.com
DocumentRoot "/www/mysite/htdocs"

# Nombre de los Hosts virtuales
NameVirtualHost 192.168.1.100
Include vhost1.domain.com.conf
Include vhost2.domain.com.conf
```

Tenga en cuenta que la directiva `Include` sigue a la directiva `NameVirtualHost`. La directiva `Include` le dice a Apache que lea los archivos `vhost1.domain.com.conf` y `vhost2.domain.com.conf` para cargar la configuración del host virtual. Estos archivos pueden tener los contenidos `<VirtualHost>` que normalmente pone en `httpd.conf`. El beneficio de esta aproximación es que puede crear una configuración muy elaborada para cada host virtual sin complicar el archivo `httpd.conf`.

# Configuración automática de host virtuales utilizando mod\_perl

Aunque el uso de la directiva `Include` (discutida en la sección anterior) ayuda a simplificar `httpd.conf`, sigue sin ser una solución lo suficientemente buena para sitios con muchos servidores Web, que ejecutan muchos sitios Web virtuales. Lo ideal es que escriba la configuración de Apache con un programa o un script, si es posible, de modo que se creen todas esas opciones de configuración habituales de forma automática.

Si utiliza el módulo `mod_perl` con Apache (capítulo 16), puede escribir código Perl para generar su configuración Apache. Como la compilación y la instalación de `mod_perl` se realiza en el capítulo 16, no lo vamos a repetir aquí.

Podrán acudir a capítulos siguientes, en caso de que necesite tener compilado e instalado soporte `mod_perl` en Apache. Recuerde que cuando compila `mod_perl` siguiendo las instrucciones del capítulo 16, tiene que utilizar la opción `EVERYTHING=1` o la opción `PERL_SECTIONS=1` con el script de configuración. Para utilizar código Perl para generar la configuración Apache, necesita utilizar el contenedor `<Perl>` en `httpd.conf`. Por ejemplo:

```
<Perl>

#
# Aquí se coloca su código Perl
#

1;
</Perl>
```

Este contenedor Perl es el mínimo absoluto que debe tener en `httpd.conf` para ser capaz de utilizar la configuración basada en Perl. La última línea dentro del contenedor devuelve 1 (valor verdadero), que es el valor necesario para satisfacer `mod_perl`. Su código puede ser cualquier script de Perl. El código en un contenedor `<Perl>` se compila en un paquete especial y `mod_perl` comunica la información de configuración al módulo de configuración general de Apache. La sintaxis que describe la configuración, se discute más tarde.

Las directivas que toman un solo valor se representan con variables escalares (concepto de Perl). Por ejemplo:

```
User httpd
```

Esta directiva `User` toma una sola cadena como valor y, por lo tanto, se puede escribir del siguiente modo:

```
<Perl>
$User = "httpd";
```

```
1;  
</Perl>
```

A continuación tenemos un ejemplo de configuración:

```
<Perl>  
    $User = "httpd";  
    $Group = "httpd";  
    $ServerAdmin      = 'kabir@mobidac.com';  
    $MinSpareServers = 5;  
    $MaxSpareServers = 5;  
    $MaxClients = 40;  
    1;  
</Perl>
```

Las directivas que necesitan varios valores, se pueden representar como listas. Por ejemplo, PerlModule Apache::TestOne Apache::TestTwo, se puede representar del siguiente modo:

```
@PerlModule = qw(Apache::TestOne Apache::TestTwo );
```

Los contenedores se representan utilizando un hash (es una construcción de programación de ordenadores que se utiliza mucho en Perl y en otros lenguajes modernos como C). Por ejemplo:

```
<VirtualHost 206.171.50.50>  
    ServerName www.nitec.com  
    ServerAdmin kabir@nitec.com  
</VirtualHost>
```

se puede representar del siguiente modo:

```
$VirtualHost{"206.171.60.60"} = {  
    ServerName => 'www.nitec.com',  
    ServerAdmin => 'kabir@nitec.com'  
}
```

Un ejemplo ligeramente más complicado sería el siguiente:

```
$Location{"/some_dir_alias/"} = {  
    AuthUserFile => '/www/nitec/secret/htpasswd',  
    AuthType => 'Basic',  
    AuthName => 'Subscribers Only Access',  
    DirectoryIndex => [qw(welcome.html welcome.htm)],  
    Limit => {  
        METHODS => POST GET',  
        require => 'user reader'  
    },  
};
```

En este ejemplo, el segmento de configuración se utiliza para crear un área restringida, utilizando el método de autentificación HTTP básica, para un alias

llamado /some dir alias/. Además, se define un conjunto de nombres de archivo como índices de directorio para este alias.

Puede definir otros contenedores como <Directory>, <Files>, de esta misma forma. A continuación se presentan algunos ejemplos de Web, host\_a, host\_b y host\_c, y se determina que el host\_a es más poderoso que el host\_b, y el host\_b más que el host\_c. Siga este ejemplo del archivo httpd.conf, para definir un contenedor <Perl> que le permita crear una sola configuración para los tres host:

```
<Perl>
# Obtiene el nombre del host utilizando la herramienta Unix
hostname
# y lo almacena en la variable $thisHost.
my $thisHost = '/bin/hostname';
if ($thisHost =~ /host_a/) {
    # la configuración de host_a se coloca aquí
    $MinSpareServers = 10;
    $MaxSpareServers = 20;
    $StartServers = 30;
    $MaxClients = 256;
}
elsif ($thisHost =~ /host_b/) {
    # la configuración de host_b se coloca aquí
    $MinSpareServers = 5;
    $MaxSpareServers = 10;
    $StartServers = 10;
    $MaxClients = 50;
}
else {
    # la configuración de host_c se coloca aquí
    $MinSpareServers = 3;
    $MaxSpareServers = 5;
    $StartServers = 5;
    $MaxClients = 30;
}
1;
</Perl>
```

Para hacer este escenario más interesante, supongamos que tiene distintos host virtuales para cada uno de los tres host y que le gustaría configurarlos de un modo elegante. Por ejemplo:

```
<Perl>
# Obtiene el nombre del host utilizando la herramienta Unix
hostname
# y lo almacena en la variable $thisHost.

my $thisHost = '/bin/hostname';
my $thisDomain = 'mydomain.com';
my @vHosts = ();
```

```

my $anyHost;

if ($thisHost =~ /(host_a)/) {

    # la configuración de host_a se coloca aquí
    @vHosts = qw(gaia, athena, romeo, juliet, shazam);

} elsif ($thisHost =~ /host_b/) {

    # la configuración de host_b se coloca aquí
    @vHosts = qw(catbart, ratbart, dilbert);

} else {

    # la configuración de host_c se coloca aquí
    @vHosts = qw(lonelyhost);

}

for $anyHost (@vHosts) {
    %{$VirtualHost{"$anyHost.$domainName"}} = {
        "ServerName" => "$anyHost.$domainName",
        "ServerAdmin" => "webmaster@$anyHost.$domainName"
    }
}

1;

</Perl>

```

Una vez que ha creado una configuración basada en Perl adecuada para sus servidores Apache, puede comprobar su sintaxis en el código para asegurarse de que el código es correcto sintácticamente, ejecutando el comando `/usr/local/apache/bin/apachectl configtest`. Si hay un error de sintaxis, verá un mensaje de error en la pantalla. Corrija el error y vuelva a utilizar este comando para asegurarse de que Apache acepta el código.

## Generar la configuración de host virtuales utilizando el script makesite

Si añadir host virtuales se ha convertido para usted en un problema diario o semanal, debido a que trabaja para un gran ISP u organización, cuyo departamento de marketing decide crear nuevos sitios Web con frecuencia, necesita el makesite. Se trata de un sencillo script de Perl que escribí hace años y que uso utilizando para crear host virtuales. Este script se encuentra en el CD-ROM y, además, se puede bajar de <http://sourceforge.net/projects/mkweb/>.

Por ejemplo, para crear un nuevo host virtual llamado `newsite.com`, puede ejecutar `makesite newsite.com` y el script crea y adjunta la configuración `httpd.conf` necesaria. Además, creará los archivos de configuración DNS necesarios, utilizando dos plantillas. Para utilizar `makesite`, siga estos pasos.

1. Copie los script `makesite` y los archivos `named.template` y `httpd.template`, en una localización apropiada en su servidor. Normalmente yo guardo el `makesite` en el directorio `/usr/bin`, por lo que se encuentra en mi ruta normal. Recomiendo que cree el directorio `/var/makesite` y que sitúe los archivos `named.template` y `httpd.template` en ese directorio. En los siguientes pasos, supongo que lo ha hecho.
2. Utilice su editor de texto preferido y modifique el script `makesite`. Tíene que modificar una o más líneas de las siguientes:

```
my $MAKESITE_DIR      = '/var/makesite';
my $USER               = 'httpd';
my $GROUP              = 'httpd';
my $PERMISSION         = '2770';
my $BASE_DIR           = '/www';
my $HTDOCS             = 'htdocs';
my $CGIBIN              = 'cgi-bin';
my $NAMED_PATH          = '/var/named';
my $NAMED_FILE_EXT     = '.zone';
my $NAMED_TEMPLATE      = "$MAKESITE_DIR/named.template";
my $NAMED_CONF          = '/etc/named.conf';
my $HTTPD_CONF          = '/usr/local/apache/conf/httpd.conf';
my $VHOST_TEMPLATE      = "$MAKESITE_DIR/httpd.template";
my $LOG_FILE             = "$BASE_DIR/makesite.log";
```

Puede necesitar realizar los cambios siguientes:

- Si siguió mi recomendación y creó `/var/makesite` para guardar los archivos `named.template` y `httpd.template`, no necesita realizar ningún cambio en `$MAKESITE_DIR`.
- Si ejecuta Apache utilizando un usuario/grupo distinto de `httpd`, cambie los valores de `$USER` y `$GROUP`. Si quiere que los permisos por defecto del directorio Web sean distintos de 2770, entonces cambie el valor `$PERMISSION`.
- Si ejecuta Apache utilizando un usuario/grupo distinto de `httpd`, cambie los valores de `$USER` y `$GROUP`. Si quiere que los permisos por defecto del directorio Web sean distintos de 2770, entonces cambie el valor `$PERMISSION`.
- Si quiere que los directorios de los documentos raíz sean distintos de `htdocs`, cambie `$HTDOCS`.

- Del mismo modo, si no quiere utilizar el alias tradicional /cgi-bin/ para ScriptAlias, cambie \$CGIBIN.
  - En los sistemas Linux, el directorio de registros DNS por defecto es /var/named; si lo ha cambiado de ruta, asegúrese que cambia \$NAMED\_PATH.
  - Si mantiene httpd.conf en una ruta distinta a /usr/local/apache, cambie el valor de \$HTTPD\_CONF.
3. Modifique /var/makesite/named.template para que refleje el nombre del servidor y los nombres de los servidores Web para su sistema. El archivo por defecto named.template supone que los nombres de sus servidores principal y secundario son ns1.domain.com y ns2.domain.com; que el nombre de su servidor de correo es mail.domain.com; y que su servidor Web es www.domain.com. Asegúrese de cambiar estos nombres de acuerdo con su configuración.
4. Asegúrese de que /var/makesite/httpd.template tiene todas las directivas que quiere añadir a cada nuevo host virtual que ha creado utilizando este script.

Ahora está preparado para ejecutar el script con el fin de crear un nuevo host virtual. Como precaución, haga una copia de seguridad de los archivos /etc/named.conf y /usr/local/apache/httpd.conf. Para crear un nuevo host virtual llamado vhost1.com ejecute el comando makesite vhost1.com.

Examine el archivo /usr/local/httpd.conf; debería ver:

```

#
# Configuración de www.vhost1.com
#
<VirtualHost www.vhost1.com>
  ServerName www.vhost1.com
  ServerAdmin webmaster@vhost1.com

  DocumentRoot /tmp/vhost1/htdocs
  ScriptAlias /cgi-bin/ /tmp/vhost1/cgi-bin/

  ErrorLog logs/www.vhost1.com.error.log
  TransferLog logs/www.vhost1.com.access.log

</VirtualHost>

#
# Final de la configuración de www.vhost1.com
#

```

Esta es la configuración <VirtualHost> que crea el script makesite.

**TRUCO:** Si está creando nombre de host virtuales por primera vez, asegúrese de que añade NameVirtualHost IP\_Address como última línea en el archivo httpd.conf antes de ejecutar el script makesite por primera vez.

Además, compruebe el archivo /etc/named.conf; debería ver lo siguiente:

```
// vhost1.com was created on 2001-04-25-17-25
zone "vhost1.com" {
    type master;
    file "vhost1.zone";
};
```

Como puede ver, el script crea la información de configuración de la zona apropiada para el host virtual.

Puede encontrar la información sobre la zona DNS en el archivo /var/named/vhost1.zone, que muestra:

```
@ IN SOA vhost1.com. hostmaster.vhost1.com. (
    20010425000 ; serial YYYYMMDDXXX
    7200         ; refresh
    3600         ; (1 hour) retry
    1728000      ; (20 days) expire
    3600)        ; (1 hour) minimal TTL

; Nombre de los servidores
IN NS ns1.domain.com.
IN NS ns2.domain.com.
IN MX 10 mail.domain.com.

; registros CNAME
www IN CNAME www.domain.com.
```

Mire en /www (o en cualquier \$BASE\_DIR); debería ver el directorio vhost1 con los archivos y los subdirectorios siguientes:

```
./vhost1
./vhost1/htdocs
./vhost1/htdocs/index.html
./vhost1/cgi-bin
```

Reinicie el servidor utilizando el comando killall -HUP named. Además, reinicie el servidor Apache utilizando el comando /usr/local/apache/bin/apachectl restart. Debería ser capaz de acceder al nuevo sitio Web virtual utilizando www.vhost1.com/.

La página por defecto index.html se encuentra allí para ayudarle a identificar el dominio en la Web.

# Gestionar host virtuales utilizando MySQL con el módulo mod\_v2h

El módulo mod\_v2h es un módulo de hospedaje en masa de host con soporte para realizar traducción de rutas URL desde la base de datos MySQL. Este módulo, puede cachear traducciones de rutas URL en la memoria para aumentar la velocidad. Necesitará tener instalado MySQL en un servidor. Para aprender algo sobre MySQL visite [www.mysql.com](http://www.mysql.com).

Con MySQL instalado, siga los pasos siguientes para compilar e instalar mod\_v2h:

1. Baje la última versión de mod\_v2h de [www.fractal.net/mod\\_v2h.tm](http://www.fractal.net/mod_v2h.tm).
2. Como raíz, extraiga la distribución de la fuente utilizando el comando tar xvzf mod\_v2h.tar.gz en el subdirectorio de módulos del árbol fuente de Apache. Por ejemplo, si mantiene la fuente Apache en el directorio /usr/local/src/httpd\_2\_0\_16, entonces extraiga el archivo mod\_v2h.tar.gz al directorio /usr/local/src/httpd\_2\_0\_16/modules. Se crea un subdirectorio nuevo llamado mod\_v2h.
3. Cambie su directorio a /usr/local/src/httpd\_2\_0\_16/modules/v2h y edite el archivo config.m4.

**NOTA:** Únicamente tiene que editar este archivo si nota que las rutas de los directorios include y lib para los archivos MySQL son incorrectas. Por ejemplo, en mi sistema Linux, los archivos include MySQL están instalados en el directorio /usr/include/mysql y los archivos library están en el directorio /usr/lib/mysql. El valor por defecto config.m4 señala a /usr/local/include/mysql y a /usr/local/lib/mysql para los archivos include y library, respectivamente; por lo que, tuve que corregir la ruta. Edite las rutas si es necesario. Yo tuve que añadir también -lz en la línea LDFLAGS por lo que quedó del siguiente modo LDFLAGS="\$LDFLAGS -L/usr/lib/mysql -lz -Wl,-R,/usr/lib/mysql".

4. Cambie el directorio a /usr/local/src/httpd\_2\_0\_16 y ejecute el comando autoconf para crear el archivo de configuración necesario.
5. Ejecute ./configure con cualquiera de las opciones que necesite, tal y como se muestra en el capítulo 2, y entonces ejecute los comandos de instalación make && make para compilar e instalar Apache con soporte mod\_v2h. Por ejemplo, yo ejecuté ./configure --prefix=/usr/

local/httpd --disable-module=cgi para configurar la fuente Apache y entonces ejecuté los comandos de instalación make && make para instalar Apache en /usr/local/httpd con soporte mod\_v2h y sin soporte mod\_cgi. Puede utilizar cualquiera de las opciones que ha utilizado antes, ejecutando ./config.status en lugar de ./configure.

Una vez que ha compilado Apache con mod\_v2h, tiene que utilizar las directivas mod\_v2h, que se muestran en la tabla 6.2, en httpd.conf.

**Tabla 6.2.** Las directivas de mod\_v2h

Directiva	Función
v2h	Asigna el valor On para activar este módulo. En caso contrario asigna el valor Off.
v2h_Mysql_Db	Asigna el nombre de la base de datos MySQL.
v2h_Mysql_Tbl	Asigna el nombre de la tabla de la base de datos MySQL.
v2h_Mysql_Serv_Fld	Asigna el nombre del campo de la tabla en la que está almacenado el nombre del servidor (por ejemplo, www.domain.com).
v2h_Mysql_Path_Fld	Especifica la ruta física del URI al que ha de ser traducido (es decir, /htdocs/www.fractal.net/).
v2h_Mysql_Host	Asigna el nombre del host del servidor MySQL.
v2h_Mysql_Port	Asigna el número de puerto de MySQL que se está utilizando para conectarse.
v2h_Mysql_Pass	Asigna la contraseña (si hay) para el acceso a la base de datos.
v2h_Mysql_User	Asigna el nombre de usuario (si hay) para el acceso a la base de datos.
v2h_Mysql_Env_Fld	Asigna el valor de un campo adicional de la base de datos que se utiliza para asignar una variable de entorno llamada VHE_EXTRA.
v2h_PathHead	Asigna la ruta extra que puede ser prefijada por la ruta que señala a v2h_Mysql_Path_Fld.
v2h_UseImage	Fija el valor On o el valor Off para permitir o desactivar el caching en la memoria.
v2hImagePath	Fija la ruta para almacenar la imagen de memoria.
v2h_DeclineURI	Fija el URI que será rechazado.



# 7

# Autentificación y autorización de visitantes al sitio Web

---

## En este capítulo

1. Autentificamos usuarios utilizando nombres de usuario y contraseñas.
2. Autorizamos el acceso utilizando el nombre del host o la dirección IP.
3. Autentificamos con RDBM.
4. Autentificamos utilizando la base de datos MySQL.
5. Autentificamos utilizando el archivo de contraseñas `/etc/passwd`.
6. Autentificamos utilizando cookies.

Se ha estado dando vueltas mucho tiempo al soporte para la autentificación HTTP básica en Apache. Se han escrito muchos módulos para proporcionar autentificación básica HTTP Apache. En este capítulo, vamos a ver los distintos tipos de autentificación y la forma de autorizar el acceso al servidor Web, cómo autentificar usuarios utilizando archivos de contraseña, servidores de bases de datos, etc., y cómo controlar el acceso restringiendo el acceso mediante direcciones IP o nombres de usuario.

# Autentificación vs. autorización

Hay mucha gente que confunde autentificación y autorización, y algunos, incluso, piensan que se trata del mismo concepto y no es cierto. Para entender la diferencia, considere el siguiente ejemplo. Cuando quiere visitar otro país necesita un pasaporte y una Visa. El pasaporte es un documento que le autentifica en otro país. Confirma que somos quienes decimos ser. Por lo tanto, cuando presenta su pasaporte, se está identificando ante, por ejemplo, un policía. A continuación, debe probar que está capacitado (es decir, tiene permiso) para entrar en ese país. El documento que se lo permite es la Visa. Ahora, en términos de computación, la autentificación normalmente implica la introducción de un nombre y una contraseña. Una entrada y una aceptación con éxito, exigen que seamos quien decimos ser. En otras palabras, tenemos que autenticarnos.

Para acceder a un recurso determinado va a necesitar autorización y autentificación. Por ejemplo, si está accediendo a un ordenador a las 4 de la mañana, el ordenador puede no permitirle el acceso a esa hora porque el administrador del sistema lo ha decidido así. De igual modo, podría estar autorizado para ver un sitio Web restringido desde la oficina pero no desde casa porque la política de la compañía dicta al administrador de red que todos los accesos restringidos sean realizados en el propio local.

## Entender cómo funciona la autentificación

La autentificación HTTP básica es realmente muy sencilla. Se utiliza un mecanismo de intento y respuesta para autenticar usuarios. Los pasos se muestran en la figura 7.1 y se discuten a continuación:

1. La autentificación comienza cuando un navegador Web solicita una URL que está protegida por un esquema de autentificación HTTP. Esto se muestra en la figura con el número (1).
2. El servidor Web devuelve entonces una cabecera de estado 401 junto con una cabecera WWW-Authenticate, que implica que se requiere autentificación para acceder a la URL. La cabecera contiene el esquema de autentificación que se está utilizando (actualmente sólo se soporta la autentificación HTTP básica) y el nombre real. Esto se muestra en la figura con el número (2).
3. En este momento, aparece una caja de diálogo en el navegador Web, pidiéndole al usuario que introduzca un nombre de usuario y una contraseña. Esto se muestra en la figura con el número (3).
4. El usuario introduce el nombre de usuario y la contraseña y hace clic en OK. El navegador envía entonces el nombre de usuario y la contraseña

junto con la URL solicitada al servidor. El servidor comprueba si son válidos el nombre de usuario y la contraseña. Esto se muestra en la figura con el número (4).

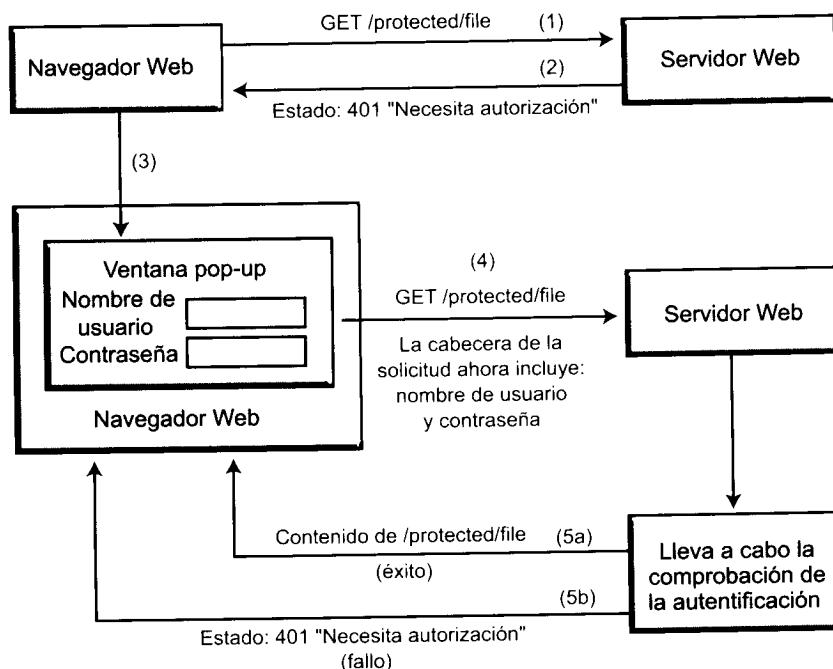


Figura 7.1. El proceso básico de autentificación HTTP

**ADVERTENCIA:** Cuando se envía la contraseña desde el sistema cliente (el host que se ejecuta en el navegador Web), no se envía como texto sino encriptado. Por el contrario, se envía como un archivo uuencoded y se transmite por Internet. Esta es la desventaja de este método de autentificación, porque cualquier persona que tenga un monitorizador de paquetes de datos circulantes o sniffer ya a ser capaz de recuperar el paquete IP que transporta la contraseña codificada. Como el esquema de codificación de datos uuencode es muy utilizado, también está disponible el codificador uuencode, de modo que prácticamente cualquiera puede decodificar una contraseña uuencode y utilizarla. Es cierto que el sniffer de paquetes ha de ser capaz de encontrar el paquete capaz de realizar la decodificación, pero lo cierto es que técnicamente posible. Este es el motivo por el cual, nunca debe utilizar la autentificación HTTP Basic para una aplicación crítica. Por ejemplo, no debería utilizar este tipo de autentificación para proteger secretos nacionales. Sin embargo, si está preparado para permitir el acceso Telnet o a rcp en su sistema, entonces está preparado para utilizar métodos

**de autentificación (en estos servicios), que son muy parecidos a la autenticación HTTP Basic. Si confía su servidor a una conexión Internet, abierto a intentos de entrada a Telnet por cualquiera que quiera intentarlo, entonces no tiene motivo para no confiar en este método.**

5. Si el nombre de usuario y la contraseña son válidos (es decir, auténticos), el servidor devuelve la página solicitada. Esto se muestra en la figura con el número (5a). Si el nombre de usuario y la contraseña no son válidos, el servidor responde con un estado 401 y envía la misma cabecera WWW-Authenticate al navegador. Esto se muestra con el número (5b) en la figura.
6. En cada solicitud posterior al mismo servidor durante la sesión del navegador, el navegador enviará el par nombre de usuario/ contraseña, de modo que el servidor no tiene que generar una cabecera de estado 401 para llamadas en la misma área del sitio. Por ejemplo, si la URL `http://apache.nitec.com/protected/` necesita autenticación HTTP basic, las siguientes llamadas a `http://apache.nitec.com/protected/a_page.html` y a `http://apache.nitec.com/protected/b_page` también necesitarán el nombre de usuario y la contraseña. Este es el motivo por el cual el navegador envía ambas cabeceras, la cabecera de estado 401 y la respuesta de cabecera WWW-Authenticate, antes de cualquier otro intento de autenticación, es decir, son emitidas por el servidor. Esto es más rápido y práctico que generar un intento para cada solicitud y hacer que el usuario introduzca el nombre de usuario y la contraseña varias veces.

## Autenticar usuarios mediante el módulo mod\_auth

El módulo `mod_auth` es el módulo de autenticación por defecto de Apache. Este módulo le permite autenticar usuarios cuyas credenciales están almacenadas en archivos de texto. Normalmente, se utiliza un archivo de texto que contiene un nombre de usuario y una contraseña. También puede utilizar un archivo de texto para crear grupos de usuarios, que podemos utilizar en la creación de reglas de autorización (que se discuten más tarde en este capítulo). Para un número pequeño de usuarios se recomienda que utilice la autenticación basada en `mod_auth`. A menudo, cuando un archivo de texto alcanza tan sólo unos cuantos miles de nombres de usuarios, la realización de la búsqueda cae dramáticamente. Por lo tanto, si tiene una base de usuarios muy grande, no se recomienda este módulo. Sin embargo, este método es perfecto para alrededor de unos cien usuarios.

Puede utilizar `/usr/local/apache/bin/httpd -l` para comprobar si este módulo está compilado en su binario de Apache. En caso contrario, tiene que utilizar la opción `--enable-module=auth` con el script `configure`, y recompilar e instalar su distribución Apache.

## Entender las directivas mod\_auth

El módulo `mod_auth` aporta las directivas de Apache `AuthUserFile`, `AuthGroupFile` y `AuthAuthoritative`. Vamos a ver estas directivas y algunos ejemplos en los que se utiliza este módulo.

### Directiva AuthUserFile

Esta directiva asigna el nombre del archivo de texto que contiene los nombres de usuario y las contraseñas utilizadas en la autenticación HTTP básica. Esta directiva requiere que proporcione un *fully qualified path* (nombre completo de la máquina incluido el dominio) al archivo, para poder utilizarlo.

**Sintaxis:** `AuthUserFile nombrearchivo`

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorio (`.htaccess`)

**Invalidar:** `AuthConfig`

Por ejemplo, la siguiente directiva asigna `/www/mobidac/secrets/.htpasswd` como archivo de nombre de usuario y contraseña:

```
AuthUserFile /www/mobidac/secrets/.htpasswd
```

El archivo con el nombre de usuario y la contraseña se crea normalmente utilizando una herramienta llamada `htpasswd`, que está disponible como un programa de soporte en la distribución estándar de Apache. El formato de este archivo es muy sencillo. Cada línea contiene un solo nombre de usuario y una contraseña encriptada. La contraseña está utilizando la función estándar `crypt()`.

**ADVERTENCIA:** Es importante que el archivo especificado por `AuthUserFile` resida fuera del árbol de documentos del sitio Web. Ponerlo dentro de un directorio Web accesible, podría permitirle bajarlo a alguien.

### Directiva AuthGroupFile

Esta directiva especifica el archivo de texto que hay que utilizar como la lista de grupos de usuarios para autenticación HTTP Basic. El nombre de archivo es

la ruta absoluta del archivo de grupo. Puede crear este archivo utilizando cualquier editor de texto.

**Sintaxis:** AuthGroupFile *nombrearchivo*

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorio (*.htaccess*)

**Invalidar:** AuthConfig

El formato de este archivo es el siguiente:

*nombregrupo:* *nombreusuario* *nombreusuario* *nombreusuario* [...]

Por ejemplo:

startrek: kirk spock picard data

Esta línea crea un grupo llamado startrek, que tiene cuatro usuarios: kirk, spock, picard y data. El icono de Advertencia de la sección anterior se aplica también a esta directiva.

## Directiva AuthAuthoritative

Si está utilizando más de un esquema de autentificación para el mismo directorio, puede fijar esta directiva con el valor *off* para que cuando falle el par nombre de usuario/ contraseña en el primer esquema, lo pase al siguiente nivel (inferior).

**Sintaxis:** AuthAuthoritative On | Off

**Predefinido:** AuthAuthoritative On

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorio (*.htaccess*)

**Invalidar:** AuthConfig

Por ejemplo, si está utilizando el módulo *mod\_auth\_mysql* (que se discutirá más tarde en este capítulo) y el módulo estándar *mod\_auth*, para proporcionar servicios de autentificación, y falla el par nombre de usuario/ contraseña para uno de ellos, entonces se utiliza, si es posible, el siguiente módulo para autenticar al usuario. Cuando falla el par nombre de usuario/ contraseña en todos los módulos, el servicio vuelve a editar una cabecera de estado 401 y envía una respuesta de cabecera WWW-Authenticate para volver a realizar una autenticación. Sin embargo, si un módulo determinado autentifica con éxito un par nombre de usuario/ contraseña, los módulos de nivel inferior nunca reciben el par nombre de usuario/ contraseña.

Se recomienda que deje el valor por defecto porque no debería diseñar un esquema de autentificación gradual en el que un usuario pasa al siguiente.

## Crear una sección sólo de miembros en su sitio Web

Utilizando las directivas mod\_auth puede crear una sección sólo de miembros en su sitio Web que necesite autenticación basada en nombre de usuario/contraseña.

Por ejemplo, vamos a suponer que quiera crear una sección sólo de miembros llamada `http://your_server_name/memberonly`. A continuación se presentan los pasos que hay que seguir.

1. Tiene que determinar el directorio físico al que quiere restringir el acceso: la mayoría de la gente utiliza un directorio dentro del directorio especificado por DocumentRoot, pero puede utilizar el directorio que quiera siempre que el usuario Apache (fijado por la directiva User) tenga permiso para leer el contenido del directorio. Voy a suponer que su DocumentRoot está asignado en /www/mysite/htdocs y que quiere restringir el acceso al directorio llamado /www/mysite/htdocs/memberonly.
2. Modifique el archivo httpd.conf para crear un alias llamado /memberonly/, como se muestra a continuación:  

```
Alias /memberonly/ "/www/mysite/htdocs/memberonly/"
```
3. A continuación, añada las directivas siguientes al archivo httpd.conf para establecer /memberonly/ como una sección restringida que requiere autenticación de usuario.

```
<Location /memberonly/>
    AuthName "Member-Only Access"
    AuthType Basic
    AuthUserFile /www/secrets/.members
    require valid-user
</Location>
```

Aquí, la directiva AuthName simplemente crea una etiqueta que el navegador Web despliega a los usuarios. Esta etiqueta tiene que ser significativa para que los usuarios sepan qué es lo que se está solicitando. Asegúrese de que utiliza las dobles comillas como se muestra arriba. AuthType siempre se fija en Basic porque HTTP sólo soporta autenticación Basic por defecto. El AuthUserFile señala a un archivo de contraseñas llamado.members. La directiva Require exige que sólo tengan acceso los usuarios válidos.

4. Ahora, utilice la herramienta htpasswd para crear un archivo de contraseñas. Suponiendo que tiene instalado Apache en /usr/local/apache, debe ejecutar el comando htpasswd del modo siguiente en el caso de que sea la primera vez que lo ejecuta:

```
/usr/local/apache/bin/htpasswd -c path_to_password_file  
username
```

La opción `-c` sólo se necesita para crear el archivo y sólo se debe utilizar una vez. Por ejemplo, para crear el primer usuario llamado `mrbert` para la configuración `/memberonly/`, ejecute:

```
/usr/local/apache/bin/htpasswd -c /www/secrets/.members  
mrbert
```

5. Para comprobar que se ha creado el usuario en el archivo de contraseñas, vea su contenido utilizando el editor de texto. Además ha de estar seguro de que sólo el usuario Apache (asignelo utilizando la directiva `User`) puede acceder a este archivo. Por ejemplo, si ejecuta Apache como usuario `httpd`, puede ejecutar los comandos `chown httpd:httpd /www/secrets/.members && chmod 750 /www/secrets/.members` para permitir la lectura de este archivo únicamente al usuario `httpd` (y al grupo).
6. Reinicie el servidor Apache utilizando el comando `/usr/local/apache/bin/apachectl restart`.
7. Ahora, utilice `http://your_server_name/memberonly/` para acceder a la sección sólo de miembros; le pedirá el nombre de usuario y la contraseña. Debería ver el valor de `AuthName ("Member-Only Access")` en la caja de diálogo desplegada.
8. Introduzca un nombre de usuario y una contraseña que no sean válidos y verá un mensaje de rechazo.
9. Para terminar, intente acceder al sitio de nuevo e introduzca un nombre de usuario y una contraseña válidos haya creado por la herramienta `htpasswd`. Debería tener acceso a la sección restringida.

**NOTA:** Si está utilizando el formato de registro habitual para el acceso de registro, puede ver los nombres de usuario registrados en sus archivos de registro.

## Crear una sección sólo de miembros utilizando un archivo `.htaccess`

Cuando tenemos organizaciones del tipo Internet Service Providers (ISP) y grandes compañías con muchos departamentos ejecutando sitios Web virtuales en el mismo servidor Web, añadir configuración sólo de usuarios en `httpd.conf` (que se discutió en la última sección), es posible que no sea una buena solución,

porque tendrá que añadir o eliminar configuraciones tan rápidamente como los usuarios (en caso de un sistema ISP) soliciten esos cambios. Utilizando una autenticación basada en .htaccess, sin embargo, puede permitir a un usuario o a un departamento crear tantas secciones sólo de miembros como ellos quieran sin su intervención, una inestimable ayuda para un administrador de sistemas muy ocupado.

Para utilizar la autenticación basada en .htaccess para autenticación sólo de miembros, siga los siguientes pasos.

1. Añada la siguiente directiva en su archivo httpd.conf:

```
AccessFileName .htaccess
```

**NOTA: Si desea permitir la autenticación basada en .htaccess sólo en un host virtual, añada esta directiva dentro del contenedor <VirtualHost> apropiado.**

2. Cambie la siguiente configuración por defecto:

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
```

a:

```
<Directory />
    Options FollowSymLinks
    AllowOverride AuthConfig
</Directory>
```

Esto permite utilizar las directivas de autorización (AuthDBMGroupFile, AuthDBMUserFile, AuthGroupFile, AuthName, AuthType, AuthUserFile, Require, y similares) en un archivo .htaccess.

3. Reinicie el servidor Apache utilizando el comando /usr/local/apache/bin/apachectl.
4. Ahora puede crear un archivo .htaccess en cualquier directorio Web accesible y controlar el acceso. Necesita tener estas directivas en el archivo .htaccess:

```
AuthName "Enter Appropriate Label Here"
AuthType Basic
AuthUserFile path_to_user_password_file
Require valid-user
```

Por ejemplo, imagine que tiene un directorio, llamado /www/mysite/htdocs/asb y que quiere restringir el acceso a este directorio a los

usuarios que se encuentran en la lista /www/mysite/secrets/users.pwd. Para hacerlo, debería utilizar la siguiente configuración:

```
AuthName "ASB Member Only Access"
AuthType Basic
AuthUserFile /www/mysite/secrets/users.pwd
Require valid-user
```

**NOTA:** Asegúrese de que únicamente el usuario Apache puede leer el archivo .htaccess (hágalo utilizando la directiva User). Por ejemplo, si ejecuta Apache como httpd, entonces debería ejecutar los comandos chown httpd:httpd .htaccess && chmod 750 .htaccess del directorio en el que ha guardado el archivo. Tenga en cuenta, además, que la creación o modificación de un archivo .htaccess no necesita reiniciar el servidor Apache, por lo tanto, puede poner a prueba la sección restringida de su sitio Web para determinar si el proceso de autentificación está funcionando adecuadamente.

## Agrupar usuarios para accesos restringidos a distintas secciones Web

Si sus usuarios necesitan tener acceso a distintas partes de su sitio Web, tiene varias opciones. En lugar de una configuración de usuario válida, que abra la sección restringida para todos los usuarios válidos, puede utilizar nombres de usuarios específicos. Por ejemplo:

```
<Location /financial>
AuthName "Members Only"
AuthType Basic
AuthUserFile /www/mysite/secrets/.users.pwd
require cgodsave jolson
</Location>

<Location /sales>
AuthName "Members Only"
AuthType Basic
AuthUserFile /www/mysite/secrets/.users.pwd
require esmith jkirk
</Location>
```

En este caso, sólo tienen acceso a la sección /financial, los usuarios cgodsave y jolson, y los usuarios esmith y jkirk tienen acceso a la sección /sales. Sin embargo, nombrar a todos los usuarios en la configuración es muy engorroso y a menudo una tarea imposible. Una aproximación es crear cualquiera de los archivos separados de contraseñas, que harán que los segmentos de configuración anterior se conviertan en:

```

<Location /financial>
AuthName "Members Only"
AuthType Basic
AuthUserFile /www/mysite/secrets/.financial-team.pwd
require valid-user
</Location>

<Location /sales>
AuthName "Members Only"
AuthType Basic
AuthUserFile /www/mysite/secrets/.sales-team.pwd
require valid-user
</Location>

```

Ahora, añada únicamente los usuarios que han de añadirse a /www/mysite/secrets/.financial-team.pwd, en este caso, cgodsave y jolson, y añada únicamente los usuarios que deberían añadirse a /www/mysite/secrets/.sales-team.pwd, en este caso, esmith y jkirk.

Sin embargo, si no quiere mantener varios archivos de contraseñas, existe otra aproximación. Por ejemplo, observe los siguientes segmentos de configuración:

```

<Location /financial>
AuthName "Members Only"
AuthType Basic
AuthUserFile /www/mysite/secrets/.members.pwd
AuthGroupFile /www/mysite/secrets/.groups
require group financial
</Location>

<Location /sales>
AuthName "Members Only"
AuthType Basic
AuthUserFile /www/mysite/secrets/.members.pwd
AuthGroupFile /www/mysite/secrets/.groups
require group sales
</Location>

```

Aquí se utiliza el mismo archivo de contraseña .members.pwd para las dos localizaciones pero cada localización utiliza un grupo distinto. El archivo de grupo es común porque un archivo de grupo puede contener varios grupos. El archivo de grupo /www/mysite/secrets/.groups es un archivo de texto sencillo, que para el ejemplo anterior sería:

```

financial: cgodsave jolson
sales: esmith jkirk

```

Ahora, para añadir un nuevo usuario a un grupo no necesita cambiar el archivo httpd.conf (o si está utilizando los archivos .htaccess, los contenedores <Location>). Puede añadir simplemente un usuario al grupo apropiado en el archivo de grupo, una vez que ha creado la cuenta de usuario utilizando el comando htpasswd.

# Autorizar el acceso mediante el nombre del host o las direcciones IP

En este esquema de autorización, el control de acceso está controlado por el nombre del host o por la dirección IP del host. Cuando se solicita un recurso determinado, el servidor Web comprueba si el host solicitado tiene el acceso permitido al recurso y toma una acción basada en este hecho.

La distribución estándar de Apache incluye un módulo llamado `mod_access`, que permite el control de acceso basado en el nombre del host de Internet de un cliente Web. El nombre del host puede ser un *fully qualified domain name* (FQDN), como `blackhole.mobidac.com`, o una dirección de IP, como `192.168.1.100`. El módulo aporta el soporte de control de acceso utilizando estas directivas Apache: `allow`, `deny`, `order`, `allow from env=variable` y `deny from env=variable`.

## Directiva `allow`

Esta directiva le permite definir una lista de host (que contenga uno o más host o direcciones IP) que tienen el acceso permitido a un directorio determinado. Cuando se ha especificado más de un host o dirección IP, hay que separarlos con espacios. La tabla 7.1 muestra los valores posibles para la directiva.

**Sintaxis:** `allow from host1 host2 host3 ...`

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorio (`.htaccess`)

**Invalidar:** `Limit`

**Tabla 7.1.** Posibles valores para la directiva `Allow`

Valor	Ejemplo	Descripción
<code>All</code>	<code>allow from all</code>	Esta palabra reservada permite el acceso a todos los host. El ejemplo muestra cómo utilizar esta opción.
<b>Un FQDN de un host</b>	<code>allow from wormhole.mobidac.com</code>	Sólo se permite el acceso al host que tiene el nombre de dominio (FQDN) especificado. La directiva <code>allow</code> del ejemplo, solo sólo permite el acceso a <code>wormhole.mobidac.com</code> . Tenga en cuenta que se comparan todos los componentes; <code>toys.com</code> no coincidirá con <code>etoyos.com</code> .

Valor	Ejemplo	Descripción
<b>Un nombre parcial de dominio de un host</b>	allow from .mainoffice. mobidac.com	Únicamente los host que tienen el mismo nombre de host, tienen el acceso permitido. El ejemplo permite que todos los host de la red .mainoffice.mobidac.com accedan al sitio. Por ejemplo, developer1.mainoffice.mobidac.com y developer2.mainoffice.mobidac.com tienen acceso al sitio. Sin embargo, developer3.baoffice.mobidac.com no tiene permiso de acceso.
<b>Una dirección IP completa de un host</b>	allow from 192.168.1.100	Sólo las direcciones IP especificadas tienen el acceso permitido. El ejemplo muestra la dirección IP completa (están presentes los cuatro octetos de IP), 192.168.1.100, que tiene permiso de acceso.
<b>Una dirección IP parcial</b>	1: allow from 192.168.1 2: allow from 130.86	Cuando están presentes menos de cuatro octetos de una dirección IP en la directiva <code>allow</code> , la dirección IP parcial se estudia de izquierda a derecha, y los host que tienen el mismo patrón de dirección IP (es decir, pertenecen a la misma máscara de red), tienen el acceso permitido. En el primer ejemplo, todos los host con direcciones IP en el rango de 192.168.1.1 a 192.168.1.255 tienen acceso. En el segundo ejemplo, todos los host de la red tienen el acceso permitido.
<b>Un par red/máscara de red</b>	allow from 192.168.1.0/255.255.255.0	Esto le permite especificar un rango de direcciones IP utilizando la dirección de red y de la máscara de red. El ejemplo permite el acceso solo a los host con la dirección IP en el rango de 192.168.1.1 a 192.168.1.255.
<b>Una especificación red/nnn CIDR</b>	allow 206.171.50.0/24	Es similar a la entrada anterior, excepto en que la máscara de red consiste en nnn de alto nivel. El ejemplo es equivalente a permitir el acceso a los host con las direcciones IP desde 206.171.50.0/255.255.255.0.

## Directiva deny

Esta directiva es exactamente la contraria de la directiva `allow`. Le permite definir una lista de host que tienen el acceso denegado a un directorio específico.

fico. Al igual que la directiva `allow`, puede aceptar todos los valores de la tabla 7.1.

**Sintaxis:** `deny from host1 host2 host3 [...]`

**Contexto:** directorio, localización, archivo de control de acceso en el ámbito de directorio (`.htaccess`)

**Invalidar:** Limit

## Directiva order

Esta directiva controla el modo en que Apache evalúa tanto la directiva `allow` como la directiva `deny`.

**Sintaxis:** `order deny, allow | allow, deny | mutual-failure`

**Predefinido:** `order deny, allow`

**Contexto:** directorio, localización, archivo de control de acceso en el ámbito de directorio (`.htaccess`)

**Invalidar:** Limit

Por ejemplo, la siguiente directiva rechaza el acceso al host `myboss.mycompany.com`, mientras permite el acceso al directorio del resto de los host. El valor de la directiva `order` es una lista separada por comas, que indica qué directiva tiene prioridad:

```
<Directory /mysite/myboss/rants>
    order deny, allow
    deny from myboss.mycompany.com
    allow from all
</Directory>
```

Normalmente, la que afecta a todos los host tiene la menor prioridad. En el ejemplo anterior, como la directiva `allow` afecta a todos los host, tiene la menor prioridad.

Aunque `allow`, `deny` y `allow, deny` son los valores más utilizados para la directiva `order`, puede utilizar otro valor, `mutual-failure`, para indicar que sólo los host que aparecen en la lista `allow` tienen concedido el acceso, pero no lo tienen los que se encuentran en la lista `deny`. En todos los casos, se evalúa cada directiva `allow` y cada directiva `deny`.

## Directiva allow from env=variable

Esta directiva, una variación de la directiva `allow`, permite el acceso cuando se asigna la variable de entorno.

**Sintaxis:** allow from env=variable

**Contexto:** directorio, archivo de control de acceso en el ámbito de directorio (.htaccess)

**Invalidar:** Limit

Esto sólo es útil si está utilizando otras directivas como BrowserMatch para asignar una variable de entorno. Por ejemplo, imagine que quiere permitirle a Microsoft Internet Explorer 6, la última versión de Internet Explorer, acceder al directorio en el que almacenó algunos archivos HTML con VBScript embebido. Al igual que el resto de navegadores Web principales, como el Netscape Navigator, no soportan directamente VBScript, es preferible que no tenga usuarios de Navigator en el directorio. En este caso, puede utilizar la directiva BrowserMatch para asignar una variable de entorno cuando se detecta Internet Explorer 5.5. La directiva sería:

```
BrowserMatch "MSIE 5.5" ms_browser
```

Puede utilizar el contenedor <Directory> para especificar la directiva allow:

```
<Directory /path/to/Vbscript_directory >
  order deny,allow
  deny from all
  allow from env=ms_browser
</Directory>
```

En este caso, el servidor Apache asignará la variable de entorno `ms_browser` para todos los navegadores que proporcionen la cadena "MSIE 6" como parte del identificador del agente de usuario. La directiva `allow` permitirá el acceso sólo a los navegadores para los que está asignada la variable `ms_browser`.

## **deny from env=variable**

Esta directiva, una variación de la directiva `deny`, deniega el acceso a todos los host para los que está asignado el entorno especificado.

**Sintaxis:** deny from env=variable

**Contexto:** directorio, localización, archivo de control de acceso en el ámbito de directorio (.htaccess)

**Invalidar:** Limit

Por ejemplo, si quiere denegar el acceso a todos los host utilizando Microsoft Internet Explorer, puede utilizar la directiva BrowserMatch para asignar una variable llamada `ms_browser` cada vez que un navegador se identifique al servidor con la cadena "MSIE".

```
BrowserMatch "MSIE" ms_browser
```

Ahora puede utilizar un contenedor <Directory> para especificar la directiva deny, del siguiente modo:

```
<Directory /path/to/vbscript_directory >
    order deny,allow
    allow from all
    deny from env=ms_browser
</Directory>
```

Si está interesado en bloquear el acceso a un método determinado de solicitudes HTTP, como GET, POST o PUT, puede utilizar el contenedor <Limit> para hacerlo. Por ejemplo:

```
<Location /cgi-bin>
    <Limit POST>
        order deny,allow
        deny from all
        allow from yourdomain.com
    </Limit>
</Location>
```

Este ejemplo permite solicitudes POST en el directorio cgi-bin únicamente si realizan las solicitudes los host del dominio yourdomain.com. En otras palabras, si este sitio tiene algún formulario HTML que envía datos introducidos por el usuario mediante el método POST HTTP, únicamente los usuarios de yourdomain.com estarían capacitados para utilizar estos formularios de forma efectiva. Normalmente, las aplicaciones CGI están almacenadas en el directorio cgi-bin, y muchos sitios presentan formularios HTML que utilizan el método POST para descargar datos a las aplicaciones CGI. Utilizando la configuración de control de acceso basada en host, un sitio puede permitir a cualquiera ejecutar un script CGI, pero sólo permitirle a un sitio determinado (en este caso, yourdomain.com) enviar realmente datos de uno o más scripts CGI. Eso le aporta al acceso CGI en este tipo de sitios un ligero carácter de sólo lectura. Todo el mundo puede ejecutar aplicaciones que generen salidas sin tomar ninguna entrada de usuario, pero únicamente los usuarios de un dominio determinado pueden proporcionar una entrada.

## Combinar autenticación y autorización

La autenticación de usuario HTTP básica, soportada en mod\_auth y la autorización soportada en mod\_access se puede combinar para implementar los problemas prácticos en el control de acceso. Por ejemplo, imagine que quiere permitir el acceso a un grupo de usuarios a las secciones /aolbuddies/ de su sitio Web en el caso de que estén navegando por el sitio Web mediante una conexión AOL. A continuación tenemos la configuración que puede añadir a

`httpd.conf` una vez que reemplace la ruta y los nombres de usuario por los apropiados:

```
Alias /aolbuddies/      "/path/to/web/directory/for/aolbuddies/"

<Location /aolbuddies/>
    Deny from all
    Allow from .aol.com

    AuthName "AOL Buddies Only"
    AuthType Basic
    AuthUserFile /path/to/.myusers.pwd
    AuthGroupFile /path/to/.mygroups
    require group aolbuddies

    Satisfy all
</Location>
```

La directiva `Satisfy all` le dice a Apache que sólo permita el acceso a aquéllos que pasan tanto las pruebas de autentificación como las de autorización. Cuando un usuario AOL conecta con `http://your_server/aolbuddies/` mediante AOL, se le pide al usuario que introduzca un nombre de usuario y una contraseña. Si el usuario introduce un nombre de usuario que pertenece al grupo `aolbuddies` y la contraseña del usuario es correcta, el usuario tendrá el acceso permitido.

**NOTA:** Debe añadir todos sus compañeros AOL como usuarios en `/path/to/.myusers.pwd` y además crear un grupo llamado `aolbuddies` en `/path/to/.mygroups` que tenga una lista de todos los compañeros AOL (usuarios en `/path/to/.aol` que añadió antes) en él.

## Autentificación con bases de datos relacionales

Si ejecuta un servidor de bases de datos relacionales en su red (o incluso un servidor Web) y tiene muchos usuarios (es decir, más de 1000 usuarios) para autentificar mediante la Web, puede utilizar el servidor de bases de datos para reemplazar la autentificación basada en archivos de texto (`mod_auth`), que se discutió antes. Hay varias ventajas en la utilización de un servidor de bases de datos para muchos usuarios; las ventajas principales son:

- La autentificación `mod_auth` se convierte en lenta cuando hay muchos usuarios almacenados en un archivo de texto.

- Si permite a los usuarios cambiar sus contraseñas vía Web, utilizando una aplicación personalizada, los archivos de texto no son seguros porque debe asegurar que ese acceso de escritura en el archivo está bloqueado y desbloqueado adecuadamente. Almacenar los datos en los servidores de bases de datos elimina esta carga adicional en sus scripts y aporta mucho mayor grado de integridad de datos global.
- En una base de datos relacional, puede almacenar una gran cantidad de información sobre un usuario que se puede utilizar en sus aplicaciones Web. Por lo tanto, es una buena idea, en estos casos, centralizar su base de datos de usuarios utilizando un servidor de bases de datos como MySQL.

**NOTA:** Puede utilizar servidores de bases de datos modernos como MySQL, Postgres, DB2, Oracle y Microsoft SQL, como base de datos de usuarios. La instalación de cualquiera de estos servidores de bases de datos se aparta del objetivo de este libro. Voy a suponer que tiene instalado uno de estos servidores de bases de datos en su red o en el servidor Web.

**TRUCO:** Si tiene un sitio con muchos usuarios, debería crear un servidor dedicado para la base de datos que sea accesible en su red Web. De forma ideal, el servidor de bases de datos debería ser accesible mediante una red especializada en lugar de Internet. La mayor parte del tiempo, es mejor tener una segunda interfaz Ethernet en cada sistema del servidor y crear una LAN que sólo sea accesible a los servidores de la LAN. (Ver el capítulo 23 para obtener los detalles de cómo crear tal tipo de red.)

## **Utilizar un servidor con una base de datos MySQL para la autentificación**

MySQL es el servidor de bases de datos gratuito más utilizado en la comunidad de código fuente abierto; está disponible bajo la licencia pública GNU. Es sencillo instalarlo y establecer su servidor como un servidor de bases de datos de autentificación de usuarios. Con la plataforma Linux, puede simplemente bajar e instalar el servidor, el cliente y los paquetes RPM de desarrollo y puede estar listo para usar en cuestión de minutos. El servidor MySQL está disponible en [www.mysql.com](http://www.mysql.com).

### **Crear la base de datos de autentificación de usuarios en el servidor MySQL**

Para utilizar el servidor MySQL como base de datos de autentificación, necesita tener información al menos sobre el nombre de usuario y la contraseña en

una tabla en una base de datos. Si ya tiene una tabla en la base de datos con esa información no necesita seguir los pasos siguientes.

1. Entrar en el servidor MySQL utilizando el comando `mysql -u root -p`. Le pedirá que introduzca la contraseña raíz para la base de datos.

**ADVERTENCIA:** La contraseña raíz para el servidor MySQL debería ser siempre distinta de la contraseña raíz de su sistema Unix. Sus usos son distintos y por lo tanto deben tratarse por separado.

2. Una vez que se registra en MySQL, ejecute el comando `create database auth;` que crea una base de datos llamada `auth`.
3. Cambie su base de datos actual por la nueva base de datos `auth` utilizando el siguiente comando:  
`use auth;`
4. Ahora tiene que crear una tabla llamada `wwwusers` introduciendo las siguientes líneas en la entrada de comandos MySQL.

```
create table wwwusers (
    username varchar(40)      not null primary key,
    passwd   varchar(20)       not null
);
```

Cada fila de esta tabla consiste en tres campos: `username`, `passwd` y `groups`. El campo `username` es la clave primaria, que significa que MySQL utiliza este campo para indexar la tabla, realizando búsquedas que utilizan el nombre de usuario. Este campo está limitado en 40 caracteres. La utilización de `varchar` (carácter variable) en lugar de un carácter fijo ahorra espacio si los nombres de usuario no son siempre de 40 caracteres. El campo `username` no puede ser nulo (es decir, está vacío) porque es además la clave principal. El campo de contraseña se llama `passwd`, que tiene un máximo de 20 caracteres y es de tipo `varchar`. No puede ser nulo.

5. Ahora introduzca el comando `describe wwwusers`, que podría mostrar la salida siguiente:

```
mysql> describe wwwusers;
+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| username | varchar(40) |      | PRI |          |       |
| passwd   | varchar(20) |      |     |          |       |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Esto confirma que la tabla `wwwusers` fue creada como queríamos.

6. Ahora necesita añadir usuarios a la tabla. Para añadir usuarios de forma manual, necesita ejecutar la siguiente sentencia SQL:

```
insert into wwwusers (username, passwd)
    values ('user_name',
            'user_password'
        );
```

Por ejemplo:

```
insert into wwwusers (username, passwd)
    values ('esmith','sale007');
```

Se añade un usuario llamado esmith con la contraseña sale007 al grupo sales. Añada tantos usuarios como quiera. Ver "Gestionar usuarios y grupos en una RDBM" para obtener los detalles sobre la gestión de usuarios utilizando scripts.

7. Si piensa utilizar grupos de usuarios para la autentificación, entonces ha de crear la tabla siguiente.

```
create table wwwgroups (
    username      varchar(40),
    groupname     varchar(40)
);
```

8. Ahora introduzca el comando describe wwwgroups, que debería mostrar la siguiente salida:

```
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| username   | varchar(40) | YES  |      | NULL    |       |
| groupname  | varchar(40) | YES  |      | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Esto confirma que la tabla wwwgroups fue creada como queríamos.

9. Puede añadir usuarios que ya existen, a nuevos grupos utilizando la siguiente sentencia SQL:

```
insert into wwwgroups (username, groupname)
    values ('user_name',
            'name_of_the_group'
        );
```

Por ejemplo:

```
insert into wwwgroups (username, groupname)
    values ('kabir',
            'www_wheel'
        );
```

añade el usuario Kabir a un nuevo grupo llamado www\_wheel. Tenga en cuenta que puede añadir el mismo usuario a varios grupos.

## Conceder acceso al servidor Apache a la base de datos de autentificación de usuarios en MySQL

Al igual que los servidores RDBM modernos, MySQL utiliza autentificación basada en nombre de usuario y contraseña para permitir el acceso a la base de datos que almacena. Por lo tanto, antes de que pueda utilizar Apache con MySQL, debería crear un usuario MySQL llamado httpd, que tuviese acceso a cualquier base de datos que tenga pensado utilizar con Apache. A continuación tenemos la forma de crear un usuario MySQL para Apache:

1. Entre en el servidor MySQL utilizando el comando mysql -u root -p. Le pedirá que introduzca la contraseña raíz para la base de datos.
2. Una vez que está registrado en MySQL, puede editar una sentencia de permiso como la siguiente:

```
grant all privileges on name_of_database
    to username@hostname identified by 'user_password'
    with GRANT option;
```

Por ejemplo, para concederle al usuario httpd (es decir, el nombre de usuario especificado en la directiva User en el archivo httpd.conf) cuando utiliza la contraseña 2manysecrets, todos los privilegios a la base de datos llamada auth para ejecutar el host local se utiliza:

```
grant all privileges on auth
    to httpd@localhost identified by '2manysecrets'
    with GRANT option;
```

Esto permite al usuario httpd acceder a la base de datos auth del host local. Lo cual supone que el servidor MySQL y el servidor Apache se ejecuten en la misma máquina. Si la base de datos está en una máquina distinta, debería utilizar el nombre de host adecuado como sustituto del host local.

**ADVERTENCIA:** Si no tiene pensado añadir aplicaciones Web que necesiten escribir en la base de datos de autentificación, no conceda all privileges al usuario del servidor Web. Por ejemplo, si simplemente quiere autenticar al usuario pero nunca actualiza o elimina el usuario mediante el servidor Web (es decir, utilizando un script CGI u otra aplicación Web), entonces reemplaza todos los privilegios con select. Esto asegura que el usuario del servidor Web sólo tiene permitido realizar consultas select en la base de datos, que es equivalente al acceso de sólo lectura.

3. Introduzca el comando `flush privileges` para indicarle al servidor MySQL que recarge la tabla de permisos.
4. Salga del monitor del programa MySQL introduciendo `exit` en el prompt de `mysql>`.

Introduzcase en MySQL con el comando `mysql -u httpd -p` y proporcionando la contraseña apropiada (`2manysecrets` en este caso). Debería ser capaz de acceder a la base de datos introduciendo `use auth;` una vez en el prompt `mysql>`. Si no puede acceder a la base de datos, asegúrese de que se encuentra en el host que ha especificado en la sentencia emitida anteriormente.

## Compilar e instalar el módulo mod\_auth\_mysql

La situación de ventaja del servidor de bases de datos MySQL en la comunidad de código fuente abierto ha descansado sobre el desarrollo de un módulo de Apache llamado `mod_auth_mysql`.

Este módulo se puede utilizar como interfaz en un servidor MySQL para la autenticación. Puede bajar la última versión de este módulo en [www.mysql.com/Downloads/Contrib/](http://www.mysql.com/Downloads/Contrib/). A continuación podemos ver cómo puede compilar e instalar `mod_auth_mysql`.

1. Extraiga la distribución de la fuente de `mod_auth_mysql` en el directorio `/usr/local/src` como raíz. Cambie el nuevo directorio `mod_auth_mysql-version` y ejecute:

```
./configure --with-apache=/usr/local/src/apache_version --  
with-mysql=/usr
```

Asegúrese de cambiar el `/usr/local/src/apache_version` por la ruta de la distribución fuente de Apache y `/usr` por la ruta en la que están instaladas las cabeceras MySQL. Si tiene instalado MySQL con la configuración por defecto, los archivos de cabecera están instalados en `/usr/local/mysql` y que el `--with-mysql=/usr` suministrado es el valor correcto para este tipo de sistemas, porque el script `mod_auth_mysql` genera `/usr/local/mysql` adjuntando `/local/mysql` a `/usr`.

2. Ejecute `make`, cambie el directorio a la distribución fuente de Apache y ejecute `./config.status --activate-module=src/modules/auth_mysql/libauth_mysql.a` (si tiene compilado ya Apache) o `./configure --activate-module=src/modules/auth_mysql/libauth_mysql.a --prefix=/usr/local/apache` (si está compilando Apache por primera vez).
3. Para terminar, ejecute `make && make install` para compilar e instalar Apache con soporte `mod_auth_mysql`.

4. Reinicie el servidor Apache utilizando el comando /usr/local/apache/bin/apachectl restart.

## Autentificar usuarios utilizando el módulo mod\_auth\_mysql

Una vez que tiene compilado e instalado mod\_auth\_mysql, creada la base de datos que contiene las tablas de usuarios y grupos, y creado un usuario en el nivel de bases de datos para acceder a estas tablas, puede configurar Apache del siguiente modo:

1. Añada las siguientes líneas en httpd.conf, fuera de cualquier <VirtualHost> o de cualquier otro tipo de contenedor como <Directory>, <Location> y similares.

```
Auth MySQL Info db_hostname db_username db_password  
Auth MySQL General DB database_name
```

La primera directiva le dice a Apache cuál es el servidor de bases de datos con el que tiene que conectar y cuál es el nombre de usuario y la contraseña que tiene que utilizar. El nombre de usuario y la contraseña de la base de datos se crean en MySQL. Este par nombre de usuario- contraseña no debe coincidir con ninguna cuenta de usuario de su sistema. La segunda directiva establece la base de datos con la que se conecta. Si utiliza una sola base de datos para todas sus necesidades de autenticación, puede asignar el nombre de la base de datos aquí. De este modo evitara teclear el nombre de la base de datos en cada segmento de configuración de autenticación.

2. Para exigir autenticación para un subdirectorio de la raíz de documentos protected\_dir puede crear un contenedor <Directory> o un contenedor <Location> en httpd.conf, o puede utilizar el archivo .htaccess (suponiendo que tiene AllowOverride AuthConf asignado en httpd.conf en el servidor principal o en un host virtual adecuado) para tener el siguiente segmento de configuración:

```
AuthName "Members Only"  
AuthType Basic  
require valid-user  
  
Auth MySQL on  
Auth MySQL DB database_name  
Auth MySQL Password Table password_table_name  
Auth MySQL Username Field username_field_name  
Auth MySQL Password Field password_field_name  
  
Auth MySQL Group Table group_table_name  
Auth MySQL Group Field group_field_name  
  
Auth MySQL Empty Passwords off  
Auth MySQL Encrypted Passwords on
```

```
Auth_MySQL_Encryption_Types Crypt_DES  
Auth_MySQL_Scrambled_Passwords off  
Auth_MySQL_Authoritative on  
Auth_MySQL_Non_Persistent off
```

**NOTA:** No olvide reemplazar database\_name, password\_table\_name, username\_field\_name, password\_field\_name, group\_table\_name y group\_field\_name con la información apropiada.

- La directiva Auth\_MYSQL activa o desactiva mod\_auth\_mysql.
- La directiva Auth\_MYSQL\_DB determina el nombre de la base de datos, el cual sostiene la tabla de contraseñas determinada por Auth\_MYSQL\_Password\_Table y la tabla de grupos determinada por Auth\_MYSQL\_Group\_Table.
- Las directivas Auth\_MYSQL\_Username\_Field y Auth\_MYSQL\_Password\_Field determinan los nombres de los campos utilizados para almacenar el nombre de usuario y la contraseña en la tabla de contraseñas.
- La directiva Auth\_MYSQL\_Group\_Field determina el campo del nombre del grupo.
- Auth\_MYSQL\_Empty\_Passwords está fijada en off porque las contraseñas vacías no son apropiadas para la mayor parte de las necesidades de autentificación.
- El soporte de contraseñas encriptadas se activa utilizando Auth\_MYSQL\_Encrypted\_Passwords, y se fija el tipo de encriptación en el estilo tradicional de Unix Crypt\_DES utilizando Auth\_MYSQL\_Encryption\_Types.

**NOTA:** Aunque pueda elegir entre los tipos de encriptación Plaintext, Crypt\_DES y MySQL, nunca recomiendo la contraseña todo texto (Plaintext). El soporte de Auth\_MYSQL\_Scrambled\_Passwords se desactiva porque no es apropiado para la mayor parte de los escenarios.

- Como no es una buena idea permitir un esquema de autentificación gradual (es decir, si un esquema falla se utiliza otro para el mismo cliente), se activa la directiva Auth\_MYSQL\_Authoritative. Esto le indica a Apache que ignore el resto de los esquemas de autentificación utilizados para el mismo directorio. Si mod\_auth\_mysql no

puede permitir a un usuario acceder a un directorio restringido, Apache volverá a editar la solicitud de autentificación.

- La directiva `Auth_MySQL_Non_Persistent`, le indica a Apache que no se desconecte del servidor de la base de datos para cada solicitud de autentificación. Desconectarse para cada solicitud significaría que Apache tendría que conectarse para cada nueva solicitud de autentificación, lo que disminuiría el rendimiento. Por lo tanto, se recomienda el valor por defecto (`off`).
3. Si añade lo que hemos visto al contenedor `<Directory>` o al contenedor `<Location>` en `httpd.conf` necesita reiniciar el servidor Apache utilizando el comando `/usr/local/apache/bin/apachectl restart`. Por otro lado, si utiliza esta configuración en el archivo `.htaccess`, puede utilizarlo sin reiniciar el servidor.

## Utilizar otras bases de datos para autentificación de usuarios

Puede utilizar Postgres, IBM DB2, Oracle u otro servidor como su base de datos, en lugar de MySQL para la autentificación de usuarios con Apache. Aunque no va a encontrar un módulo Apache como `mod_auth_mysql` para su nuevo RDBM, puede utilizar el módulo Apache `::AuthDBI` para `mod_perl` (ver el capítulo 16, en el que se discute la utilización de `mod_perl`, para obtener los detalles de cómo instalar `mod_perl`) para comunicarse con su servidor de bases de datos y para llevar a cabo la autentificación de usuario. Siga los siguientes pasos:

1. Asegúrese de que están instalados todos los archivos `library` e `include`, incluidos en el paquete del servidor de bases de datos. Normalmente, esto significa que necesita instalar el Software Development Kit (SDK) de su RDBM.
2. Instale la última versión del módulo DBI utilizando el comando `perl -MCPAN -e 'install DBI'`.
3. Instale la última versión del driver apropiado de la base de datos para Perl (DBD) utilizando el comando `perl -MCPAN -e 'install DBD::database'`. Por ejemplo, para instalar el driver de la base de datos para IBM DB2 ejecutará `perl -MCPAN -e 'install DBD::db2'`.
4. Instale la última versión de Apache `::AuthDBI` utilizando el comando `perl -MCPAN -e 'install Apache::AuthDBI'`.
5. Cree una cuenta de usuario en la base de datos de Apache para conectarse con el servidor de la base de datos. Esta no es una cuenta de acceso a su

sistema. Es una cuenta en la ingeniería de la base de datos que concede permiso de acceso a la base de datos de usuario y a sus tablas.

6. Tiene que crear la base de datos auth y la tabla wwwuser discutida en la sección "Crear una base de datos de autenticación de usuarios en el servidor mysql". Además, tiene que crear uno o más usuarios de prueba, utilizando el script manage\_users.pl discutido en la sección "Gestionar usuarios y grupos en una RDBM".
7. Añada la siguiente línea en httpd.conf:

```
PerlModule Apache::AuthenDBI
```

Esta línea le indica a Apache que quiere utilizar el módulo Apache::AuthenDBI.

8. Hay que crear un alias llamado /memberonly/ para indicar el directorio al que quiere restringir el acceso utilizando la siguiente directiva Alias:

```
Alias /memberonly/ "path_to_restricted_access_directory"
```

Por ejemplo:

```
Alias /memberonly/ "/usr/local/apache/htdocs/protected/"
```

Aquí, el alias /memberonly/ está localizado en el directorio /usr/local/apache/htdocs/protected/.

9. A continuación, tiene que crear el siguiente segmento de configuración en httpd.conf:

```
<Location /memberonly/>
    AuthName "Home"
    AuthType Basic
    PerlAuthenHandler Apache::AuthenDBI
    PerlSetVar Auth_DBI_data_source dbi:mysql:database=auth
    PerlSetVar Auth_DBI_username          httpd
    PerlSetVar Auth_DBI_password          2manysecrets
    PerlSetVar Auth_DBI_pwd_table         wwwusers
    PerlSetVar Auth_DBI_uid_field        user
    PerlSetVar Auth_DBI_pwd_field        passwd
    PerlSetVar Auth_DBI_encrypted        on
    require valid-user
</Location>\
```

La siguiente lista organiza todo lo que acabamos de ver:

- La configuración anterior le indica a Apache que utilice Apache::AuthenDBI como manejador del alias /memberonly/.
- Las directiva PerlSetVar se utiliza para determinar pares clave=valor necesarios para este módulo.

La clave `Auth_DBI_data_source` determina el DSN de la base de datos que le indica al módulo la base de datos con la que tiene que conectar, y cuál es la Perl DBD que tiene que utilizar. En este caso el valor asignado hace que se conecte con una base de datos MySQL llamada `auth`. Debería asignar el driver a la RDBM que está utilizando. Por ejemplo, si está utilizando IBM DB2, su DSN debería ser `dbi:db2:database=auth`.

- Las claves `Auth_DBI_username` y `Auth_DBI_password` determinan el nombre de usuario y la contraseña de la base de datos que hay que utilizar para conectar con la base de datos nombrada (es decir, `auth`).

El nombre de la tabla de contraseñas está determinado por la clave `Auth_DBI_pwd_table`; de igual modo, las claves `Auth_DBI_uid_field` y `Auth_DBI_pwd_field` especifican los campos del nombre de usuario y la contraseña.

La clave `Auth_DBI_encrypted` tiene el valor `on`, de modo que se supone que las contraseñas almacenadas en la base de datos se encriptan utilizando el estilo tradicional de Unix en el que hay una función de encriptación por gestión llamada `crypt`.

- Para terminar, la directiva `require valid-user` le indica a Apache que sólo permita el acceso a aquellos usuarios que pasen la prueba de autentificación.
10. Puede reiniciar el servidor Apache utilizando el comando `/usr/local/apache/bin/apachectl restart` e intentar acceder al directorio `http://your_server_name/memberonly`, para ver si puede acceder al directorio con nombres de usuarios y contraseñas que no sean válidos. Si introduce un nombre de usuario y una contraseña válidos, debería ser autenticado.

## Gestionar usuarios y grupos en una RDBM

Gestionar usuarios y grupos en una base de datos a mano es muy engorroso. Por suerte, no tiene que enfrentarse a este tipo de tareas. Puede utilizar una asignación de scripts de Perl para manejar estas tareas rutinarias de forma eficaz. A continuación vemos cómo hacerlo.

1. Necesitará instalar el paquete DBI y los módulos DBD:: : database apropiados del Comprehensive Perl Archive Network (CPAN). Por ejemplo, si instala la base de datos MySQL, como usuario raíz puede instalar los módulos DBI y DBD de la línea de comando del siguiente modo:

```
perl -MCPAN -e 'install DBI'  
perl -MCPAN -e 'install DBD::mysql'
```

2. Compruebe si tiene los dos módulos CPAN llamados `HTTPD::UserAdmin` y `HTTPD::GroupAdmin` en su distribución instalada de Perl. Puede ejecutar los comandos locales `UserAdmin.pm` y `GroupAdmin.pm` para determinar si los tiene. Normalmente, estos dos módulos están instalados como parte de la distribución estándar. Por ejemplo, en mi sistema, estos módulos aparecen del siguiente modo:

```
/usr/lib/perl5/site_perl/5.6.0/HTTPD/UserAdmin.pm  
/usr/lib/perl5/site_perl/5.6.0/HTTPD/GroupAdmin.pm
```

Si no tiene uno de los módulos o no tiene ninguno, instálelos del siguiente modo: como raíz baje los módulos `HTTPD::UserAdmin` y `HTTPD::GroupAdmin` CPAN de CPAN. En un sistema Linux puede simplemente ejecutar los comandos siguientes para instalarlos:

```
perl -MCPAN -e 'install HTTPD::UserAdmin'  
perl -MCPAN -e 'install HTTPD::GroupAdmin'
```

**TRUCO:** Debería instalar el paquete `HTTPO::Tools` porque incluye los dos módulos así como otros módulos que son muy útiles para los servidores Web en general.

3. Copie el script `manage_users.pl` del CD-ROM en el directorio `/usr/bin` (ver el apéndice CD para obtener información sobre donde encontrar `/usr/bin`). Cambie el permiso para permitirle ejecutarlo. Asigne el permiso utilizando el comando `chmod 750 /usr/bin/manage_users.pl`.
4. Utilice su editor de texto favorito para modificar las siguientes líneas del script

```
my $DB_HOST          = 'localhost';  
my $DB_PORT          = '';  
my $DATABASE         = 'auth';  
my $DB_DRIVER        = 'mysql';  
my $DB_USER          = 'kabir';  
my $DB_PASSWORD      = $dbpwd;  
my $ENCRYPTION       = 'crypt';  
my $USER_TABLE       = 'wwwusers';  
my $USERNAME_FIELD   = 'username';  
my $PASSWORD_FIELD   = 'passwd';  
my $GROUP_TABLE      = 'wwwgroups';  
my $GROUP_FIELD      = 'groupname';  
my $MAXSZ_USER       = 40;
```

```
my $MAXSZ_PWD      = 20;  
my $MAXSZ_GRP      = 40;
```

Necesita determinar las variables que están a continuación, para el código anterior:

- La variable **\$DB\_HOST** debe asignarse al servidor de la base de datos a la que quiere conectarse. Si el servidor de la base de datos se encuentra en la misma máquina que el servidor Web, se puede dejar el valor por defecto 'localhost'.
- La variable **\$DB\_PORT** debe estar asignada al puerto del servidor. Por defecto, el puerto se selecciona automáticamente a no ser que tenga que utilizar un puerto no tradicional en el servidor de la base de datos para la conexión del cliente.
- La variable **\$DATABASE** debe asociarse al nombre de la base de datos. El nombre por defecto de la base de datos es 'auth' y sólo va a funcionar si ha seguido las instrucciones de secciones anteriores.
- La variable **\$DB\_DRIVER** debe asociarse al driver de la base de datos que necesita para conectar con el servidor de la base de datos. Para el servidor de la base de datos MySQL este driver se llama mysql y por lo tanto el valor por defecto sólo funcionará si está utilizando una base de datos MySQL.
- La variable **\$DB\_USER** debe asignarse al usuario al que se le ha concedido el acceso para crear, modificar o eliminar registros en las tablas especificadas en **\$DATABASE**. Ver "Conceder acceso al servidor Apache a la base de datos de autentificación de usuarios en MySQL", para saber cómo puede conceder acceso a los usuarios a la base de datos MySQL.
- **\$DB\_PASSWORD** no se almacena en el script para aumentar la seguridad. Debe proporcionar la contraseña necesaria para acceder a la base de datos utilizando la opción de la línea de comando **-dbpwd=database\_password** cada vez que ejecute el script **manage\_users.pl**. Puede utilizar código de hardware, pero recomiendo eliminar la contraseña del código de hardware una vez que lo haya hecho con el script.
- La variable **\$ENCRYPTION** puede asignarse en **none**, **crypt** (por defecto) o **MD5**. Cuando asignamos **none**, las contraseñas se almacenan en todo texto; cuando se utiliza **crypt**, las contraseñas se encriptan con el algoritmo de digestión de un único sentido utilizado en el entorno Unix tradicional; cuando se utiliza **MD5**, la contraseña se almacena como un valor de mensaje de digestión (**MD5**).

- La variable \$USER\_TABLE debe asignarse a la tabla de usuarios en su base de datos. Esta tabla debe tener el campo del nombre de usuario especificado por \$USERNAME\_FIELD y también el campo de la contraseña especificado por \$PASSWORD\_FIELD.
- La variable \$GROUP\_TABLE debería asignarse a la tabla de grupos en su base de datos. Esta tabla debe tener el campo del nombre de usuario especificado por \$USERNAME\_FIELD y además el campo de nombre de grupo especificado por \$GROUP\_FIELD.
- El tamaño máximo de \$USERNAME\_FIELD se determina utilizando el campo \$MAXSZ\_USER, que correspondería a lo que tiene que utilizar en el proceso de creación de \$USER\_TABLE. El tamaño de \$PASSWORD\_FIELD se controla de forma similar utilizando el campo \$MAXSZ\_PWD. Para terminar, el tamaño de \$GROUP\_FIELD se controla utilizando el campo \$MAXSZ\_GRP.

5. Guarde los cambios.

### **Añadir un nuevo usuario a la tabla de usuarios**

Para añadir un usuario nuevo a la tabla de usuarios ejecute el siguiente comando:

```
manage_user.pl -db=user \
               -action=add \
               -user=user_name \
               -password=user_password \
               -dbpwd=database_password
```

Por ejemplo, para añadir un usuario llamado kabir con la contraseña de usuario go#forward puede ejecutar el comando siguiente:

```
manage_user.pl -db=user \
               -action=add \
               -user=kabir \
               -password=go#forward \
               -dbpwd=mydbpwd
```

Tenga en cuenta que aquí mydbpwd es la contraseña de la base de datos necesaria para escribir en la base de datos.

### **Eliminar un usuario de la tabla de usuarios**

Para eliminar un usuario de la tabla de usuarios y de grupo ejecute el siguiente comando:

```
manage_user.pl -db=user \
               -action=del \
               -user=user_name \
```

```
-dbpwd=database_password \
-auton
```

Por ejemplo, para eliminar un usuario llamado kabir de las tablas de usuario y de grupo utilizando una contraseña de acceso a la base de datos mydbpwd, puede ejecutar el comando siguiente:

```
manage_user.pl -db=user \
-action=del \
-user=kabir \
-dbpwd=mydbpwd \
-auton
```



### **Actualizar una contraseña de usuario en la tabla de usuarios**

Para actualizar una contraseña de usuarios en la tabla de usuarios, ejecute el siguiente comando:

```
manage_user.pl -db=user \
-action=update \
-user=user_name \
-dbpwd=database_password
```

Por ejemplo, para actualizar la contraseña del usuario kabir a mksecret utilizando la contraseña mydbpwd de la base de datos, ejecute el siguiente comando:

```
manage_user.pl -db=user \
-action=update \
-user=kabir \
-dbpwd=mydbpwd
```

### **Añadir un usuario a un grupo**

Para añadir un usuario que ya existe a un grupo nuevo o a uno que ya existe, ejecute el siguiente comando:

```
manage_user.pl -db=group \
-action=add \
-user=user_name \
-group=group_name \
-dbpwd=database_password
```

Por ejemplo, para añadir un usuario llamado kabir a un grupo llamado administrators, ejecute el siguiente comando:

```
manage_user.pl -db=group \
    -action=add \
    -user=kabir \
    -group=administrators \
    -dbpwd=mydbpwd
```

Aquí, mydbpwd es la contraseña de la base de datos necesaria para escribir en la tabla de grupos.

### Eliminar un usuario de un grupo

Para eliminar un usuario de un grupo, ejecute el siguiente comando:

```
manage_user.pl -db=group \
    -action=del \
    -user=user_name \
    -group=group_name \
    -dbpwd=database_password
```

Por ejemplo, para eliminar un usuario llamado kabir de un grupo llamado administrators, ejecute:

```
manage_user.pl -db=group \
    -action=del \
    -user=kabir \
    -group=administrators \
    -dbpwd=mydbpwd
```

Aquí, mydbpwd es la contraseña de la base de datos necesaria para actualizar la tabla de grupos.

## Utilizar cookies para autenticar sesiones

Tal y como se ha mencionado antes en este capítulo, la autenticación HTTP Basic, requiere que el navegador Web pase siempre el nombre de usuario y la contraseña codificada (no encriptada) cada vez que se solicite una página que se encuentra bajo una sección restringida. Esto hace los ataques por interposición de intrusos muy sencillos. Este ataque implica un intruso interceptando paquetes entre un servidor Web y un navegador Web utilizando autenticación HTTP Basic para determinar las contraseñas.

La solución contra el ataque por interposición de intrusos, es utilizar una conexión secure socket layer (SSL) y un esquema de autenticación basada en sesión, en el que se autentifica al usuario una vez utilizando la autenticación HTTP Basic, y todas las solicitudes siguientes del recurso restringido son autorizadas utilizando una sesión segura (encriptada) de cookies, en lugar de la contraseña codificada.

Esta sección discute una solución que utiliza bases de datos MySQL (sin embargo, puede utilizar cualquier otra RDBM) y módulos mod\_perl de CPAN. A continuación tenemos el modo de implementar este tipo de solución.

1. Si no está instalado ya, instale el módulo mod\_perl tal y como se discute en el capítulo 16.
2. Instale dos módulos CPAN para Apache utilizando el siguiente comando como ruta:

```
perl -MCpan -e 'install Apache::AuthCookie'  
perl -MCpan -e 'install Apache::AuthTicket'
```

3. Una vez que tiene instalado estos módulos, necesitará crear la base de datos auth y la tabla wwwuser discutida en la sección "Crear la base de datos de autenticación de usuarios en el servidor mysql" de este capítulo. Necesitará también añadir las dos tablas siguientes.

```
CREATE TABLE tickets (  
    ticket hash CHAR(32) NOT NULL,  
    ts          INT NOT NULL,  
    PRIMARY KEY (ticket_hash)  
);  
  
CREATE TABLE ticketsecrets (  
    sec_version  BIGINT,  
    sec_data     TEXT NOT NULL  
);
```

Siga las instrucciones dadas en la sección "Crear la base de datos de autenticación de usuarios en el servidor mysql" de este capítulo, para añadir estas tablas en la base de datos auth.

4. Una vez que ha creado esas tablas, debe añadir un secreto en la tabla ticketsecrets. El modo más sencillo de añadir un secreto es registrarse en la base de datos y conectarse a la base de datos auth y emitir una sentencia insertada del siguiente modo:

```
insert into ticketsecrets (sec_version, sec_data) values  
('number', 'random_data');
```

5. Determine a qué localización del directorio Web quiere restringirle el acceso y emita una sesión de cookies para esto. En este ejemplo, he llamado a esta localización /protected. Añada la siguiente configuración a su archivo httpd.conf:

```
PerlModule Apache::AuthTicket  
PerlSetVar ProtectedTicketDB  
DBI:mysql:database=auth;host=localhost  
PerlSetVar ProtectedTicketDBUser httpd
```

```
PerlSetVar ProtectedTicketDBPassword secret1
PerlSetVar ProtectedTicketTable tickets:ticket_hash:ts
PerlSetVar ProtectedTicketUserTable wwwusers:username:passwd
PerlSetVar ProtectedTicketSecretTable
ticketsecrets:sec_data:sec_version
PerlSetVar ProtectedTicketPasswordStyle crypt
```

La siguiente lista le dice qué es lo que está ocurriendo en la configuración anterior:

- Aquí la directiva `PerlModule` le dice a Apache que quiere utilizar el módulo `Apache::AuthTicket`.
- Las directivas `PerlSetVar` se utilizan para asignar los distintos pares clave=valor que el módulo necesita.
- Las claves `ProtectedTicketDB` asignan el DSN para la base de datos.
- El valor de muestra `DBI:mysql:database=auth; host=localhost` le dice al módulo `AuthTicket` que queremos utilizar el driver de la base de datos MySQL (`mysql`) y conectar con la base de datos llamada `auth`, que reside en el host local (en la misma máquina que el servidor Web). Asegúrese de que cambia el nombre de host de forma apropiada, si está ejecutando el servidor de la base de datos MySQL en la misma máquina que el servidor Web Apache.
- Las directivas `ProtectedTicketDBUser` y `ProtectedTicketDBPassword` le dicen al módulo `AuthTicket` qué bases de datos de nombres de usuarios y de contraseñas se necesitan para acceder al servidor de bases de datos.
- Las claves `ProtectedTicketTable`, `ProtectedTicketUserTable` y `ProtectedTicketSecretTable` le dicen al módulo `AuthTicket` qué tablas de ticket y de usuarios utilizar en la base de datos y qué campos son necesarios.
- `ProtectedTicketPasswordStyle` determina el tipo de encriptado. Tiene tres opciones: estilo tradicional de Unix en el que hay una encriptación por digestión en una dirección, `Plaintext` (no recomendado), o `MD5`.

6. A continuación añada las siguientes líneas de configuración:

```
PerlSetVar ProtectedTicketExpires 30
PerlSetVar ProtectedTicketLogoutURI /protected/index.html
PerlSetVar ProtectedTicketLoginHandler /protectedlogin
PerlSetVar ProtectedTicketIdleTimeout 15
PerlSetVar ProtectedPath /
PerlSetVar ProtectedDomain .domain_name
```

```
PerlSetVar ProtectedSecure 1  
PerlSetVar ProtectedLoginScript /protectedloginform
```

La siguiente lista le dice lo que está ocurriendo en la configuración anterior:

- La clave `ProtectedTicketExpires` determina el tiempo de expiración de la sesión (ticket) en minutos.
- La llave `ProtectedTicketLogoutURI` determina la URL que se muestra tras utilizar un registro de usuario.
- `ProtectedTicketLoginHandler` determina la ruta del manejador de registros, que debe corresponder a un contenedor `<Location>`, tal y como se discute más tarde.
- `ProtectedTicketIdleTimeout` determina el número de minutos en los que una sesión puede estar parada.
- `ProtectedPath` determina la ruta de la cookie. El valor por defecto / asegura que se devuelve la cookie con todas las solicitudes. Puede restringir la cookie al área protegida simplemente cambiando / a / `protected` (o a cualquier localización que esté protegiendo).
- `ProtectedDomain` determina el nombre de dominio de la cookie. El punto inicial asegura que la cookie es enviada a todos los host Web en el mismo dominio. Por ejemplo, si lo fijamos en `.mobidac.com` permitiría ver la cookie en `web1.Mobidac.com` o en `web2.Mobidac.com`. También puede restringir la cookie a un sólo host especificando aquí el *fully qualified host name*.
- Fijando `ProtectedSecure` en 1 aseguramos que la cookie es segura.
- `ProtectedLoginScript` determina la localización para el formulario de registro, que es generado por el módulo.

7. Ahora necesita crear un contenedor `<Location>` para el directorio / `protected` del siguiente modo:

```
<Location /protected>  
    AuthType Apache::AuthTicket  
    AuthName Protected  
    PerlAuthenHandler Apache::AuthTicket->authenticate  
    PerlAuthzHandler Apache::AuthTicket->authorize  
    require valid-user  
</Location>
```

Aquí se le dice a Apache que pida las credenciales válidas del usuario, que ha de ser autenticado por el módulo `Apache::AuthTicket`.

8. Ahora necesita establecer los manejadores para la pantalla de registro, script de registro y las funciones logout del módulo del siguiente modo:

```
<Location /protectedloginform>
    AuthType Apache::AuthTicket
    AuthName Protected
    SetHandler perl-script
    PerlHandler Apache::AuthTicket->login_screen
</Location>

<Location /protectedlogin>
    AuthType Apache::AuthTicket
    AuthName Protected
    SetHandler perl-script
    PerlHandler Apache::AuthTicket->login
</Location>

<Location /protected/logout>
    AuthType Apache::AuthTicket
    AuthName Protected
    SetHandler perl-script
    PerlHandler Apache::AuthTicket->logout
</Location> </Location>
```

9. Una vez que ha creado la configuración anterior, asegúrese de que ha añadido al menos un usuario a la tabla wwwusers. Ver la sección "Gestionar usuarios y grupos en una RDBM" de este capítulo para obtener los detalles de cómo gestionar usuarios en una base de datos.
10. Reinicie el servidor Web de Apache utilizando el comando `/usr/local/apache/bin/apachectl restart`.
11. Asegúrese de que ve la cookie, determine en su navegador Web que pida las cookies. Para Netscape Navigator, puede comprobar el Warn me antes de almacenar una opción cookie utilizando la opción Edit>Preference>Advanced>Cookies. Para Microsoft IE, debe utilizar las opciones Tools>Internet Options>Security>Custom Levels>Cookies>Prompt.
12. Ahora acceda al directorio `http://your_server_name/protected/` y verá un formulario Web solicitando su nombre de usuario y su contraseña. Introduzca un nombre de usuario válido y una contraseña no válida y el formulario Web debería simplemente volver a mostrarse. Ahora introduzca un par nombre de usuario/contraseña válido y su navegador Web le pedirá permiso para almacenar la cookie. A continuación tenemos una muestra de sesión (ticket) inválida.

```
Cookie Name: Apache::AuthTicket_Protected
Cookie Domain: nitec.com
Path: /
Expires: End of session
```

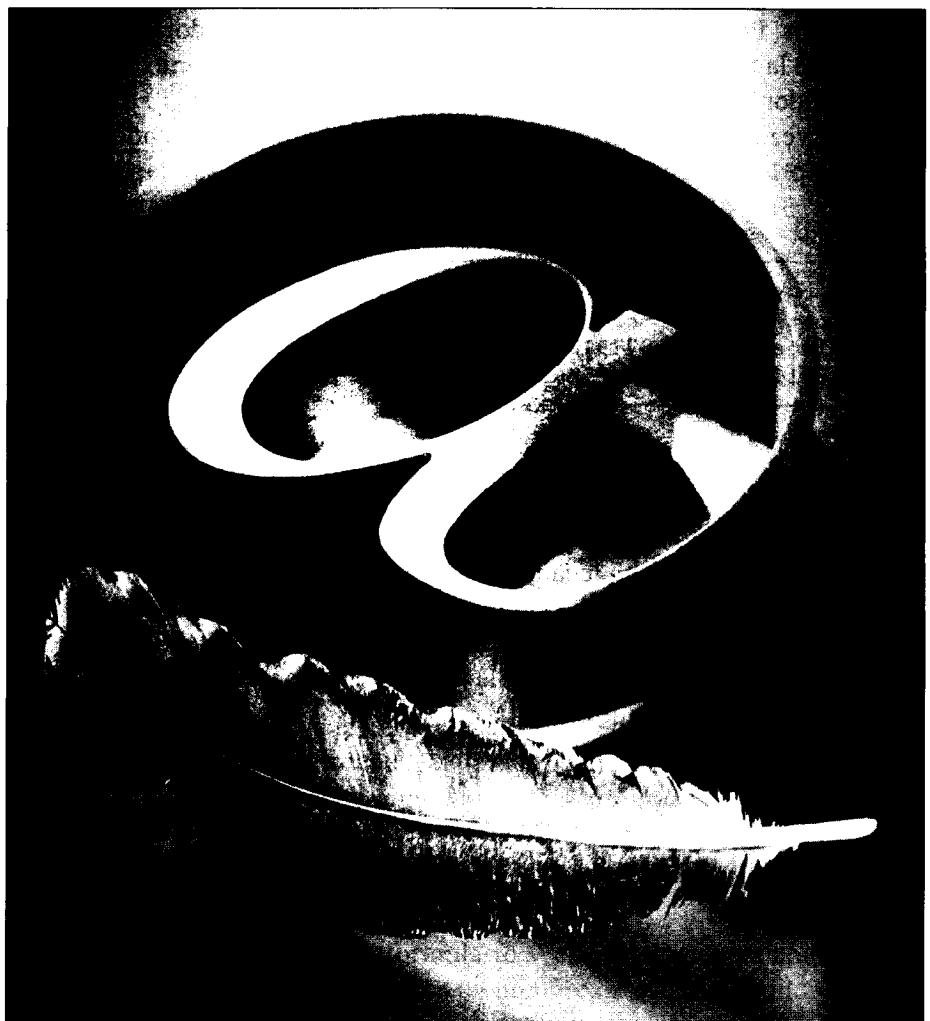
```
Secure: Yes
Data:
expires:988390493:version::user:kabir2:hash:
bf5ac94173071cde94489ef79f24b158:time:988389593
```

13. Capacite al navegador web para que almacene la cookie y tendrá acceso a la sección Web restringida.
14. A continuación, debería verificar que no hay un nuevo ticket en la tabla de tickets. Puede registrarse en su base de datos del servidor y ver el contenido de la tabla de tickets. Por ejemplo, en un sistema Linux ejecutando un servidor MySQL, puede ejecutar el comando `select * from tickets`, una vez que está registrado en MySQL mediante el comando `mysql -u httpd -p auth`. A continuación se muestra una salida:

```
mysql> select * from tickets;
+-----+-----+
| ticket_hash | ts      |
+-----+-----+
| 145e12ad47da87791ace99036e35357d | 988393278 |
| 6e115d1679b8a78f9b0a6f92898e1cd6 | 988393401 |
+-----+-----+
2 rows in set (0.00 sec)
```

Aquí MySQL informa que hay dos sesiones conectadas actualmente a un servidor Web.

15. Puede forzar a los navegadores Web para que se registren de nuevo, eliminando los tickets almacenados en esta tabla. Por ejemplo, editando el comando `delete from tickets` en su servidor de bases de datos, elimina todos los registros en la tabla de tickets y fuerza a todo el mundo a que se registre de nuevo.



# 8

# Monitorización del acceso a Apache

---

## En este capítulo

1. Monitorizamos el estado de Apache.
2. Permitimos el registro.
3. Personalizamos el registro.
4. Archivamos sus registros.
5. Realizamos un seguimiento de usuarios.
6. Analizamos sus archivos de registro.
7. Mantenemos sus archivos de registro.

Seguramente se ha preguntado alguna vez quién accede a su sitio Web o cómo funciona el servidor Apache en su sistema. La monitorización, el registro y el análisis del servidor Apache, pueden proporcionar una gran cantidad de información vital para el administrador de sistemas del servidor Web, y además puede ser de gran ayuda en los aspectos de marketing de su sitio Web. En este capítulo, le mostraré la forma de controlar y registrar información en un servidor Apache, para satisfacer sus necesidades de conocimiento.

Entre otras cosas, en este capítulo le voy a mostrar cómo:

- Acceder rápidamente a las configuraciones de los servidores Apache
- Monitorizar el estado de un servidor de Apache que se está ejecutando
- Crear archivos de registro tanto en formato CLF como en formatos personalizados
- Analizar archivos de registro utilizando aplicaciones de terceras partes

## Monitorizar Apache

Apache le permite monitorizar estos dos tipos de información valiosa vía Web:

- **Información de la configuración del servidor:** esta información es estática, pero acceder a ella rápidamente puede resultar muy útil cuando quiere determinar qué módulos están instalados en el servidor.
- **Información del estado del servidor:** esta información cambia constantemente. Puede monitorizar información del tipo tiempo de operación del servidor, número total de solicitudes servidas, transferencia total de datos, estado de los procesos hijo y manejo de los recursos del sistema.

Voy a analizar ambos tipos de información en las siguientes secciones.

## Acceder a la información de configuración con mod\_info

Se puede acceder a la información de la configuración del sistema mediante el módulo `mod_info`. Este módulo proporciona un resumen de la configuración del servidor, incluyendo todos los módulos instalados y todas las directivas de los archivos de configuración. Este módulo se encuentra en el archivo `mod_info.c`. No está compilado en el servidor por defecto. Tiene que compilarlo utilizando la opción `--enable-info` en el script de configuración. Por ejemplo:

```
./configure --prefix=/usr/local/apache \
            --with-mpm=prefork \
            --enable-info
```

Este comando, configura Apache de modo que se queda instalado en el directorio `/usr/local/apache`, configura la fuente para que se ejecute como un servidor "preforking", y habilita el módulo `mod_info`. Ejecute `make` y `make install` para compilar e instalar el nuevo servidor construido de Apache.

Una vez que tiene instalado este módulo en el servidor, puede ver la información de la configuración de la Web, añadiendo la siguiente configuración al archivo `httpd.conf`:

```
<Location /server-info>
    SetHandler server-info
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1 .domain.com
</Location>
```

Esto le permite al host local (127.0.0.1) y a cada host de su dominio, acceder a la información del servidor. No olvide reemplazar .domain.com con el nombre de dominio de su máximo nivel. Por ejemplo, si su sitio es www.nitec.com, necesita añadir:

```
Allow from 127.0.0.1 .nitec.com
```

El punto delante del nombre de dominio permite que cualquier host del dominio acceda a la información del servidor. Sin embargo, si desea limitar todo esto a un solo dominio llamado sysadmin.domain.com, cambie la línea Allow from por:

```
Allow from 127.0.0.1 sysadmin.domain.com
```

Una vez que el servidor está configurado y reiniciado, la información del servidor se obtiene del host local (es decir, ejecutando el navegador Web con Lynx en el propio servidor) accediendo `http://localhost/server-info`.

Esto devuelve una página completa de configuración del servidor y de todos los módulos. Si quiere acceder desde localizaciones distintas, utilice el nombre completo del servidor, incluido el dominio, el *fully qualified name* en lugar del nombre del host local. Por ejemplo, si su servidor Web se llama www.nitec.com, accederá a la información del servidor utilizando: `http://www.nitec.com/server-info`.

El módulo mod\_info también proporciona una directiva llamada AddModuleInfo, que le permite añadir texto descriptivo en la lista de módulos proporcionada por el módulo mod\_info. El texto descriptivo podría ser cualquier cosa incluido texto HTML. AddModuleInfo tiene la siguiente sintaxis:

```
AddModuleInfo module_name descriptive_text
```

Por ejemplo:

```
AddModuleInfo mod_info.c 'See <a href="http://localhost/manual/mod/mod_info.html">man mod_info</a>'
```

Esto muestra un enlace HTML junto con la lista de mod\_info.c, proporcionando un modo sencillo de obtener información en el módulo desde el manual online de Apache, tal y como se muestra a continuación.

```
Module Name: mod_info.c
Content handlers: (code broken)
Configuration Phase Participation: Create Server Config, Merge
Server Configs
```

```
Module Directives:  
AddModuleInfo - a module name and additional information on  
that module  
Current Configuration:  
AddModuleInfo mod_info.c 'man mod_info'  
  
Additional Information:  
man mod_info
```

También puede limitar la información desplegada en la pantalla del siguiente modo:

- **Sólo configuración del servidor.** Utilice `http://server/server-info?server`, que muestra la siguiente información:

```
Server Version: Apache/2.0.14 (Unix)  
Server Built: Mar 14 2001 12:12:28  
API Version: 20010224:1  
Hostname/port: rhat.nitec.com:80  
Timeouts: connection: 300      keep-alive: 15  
MPM Information: Max Daemons: 20 Threaded: no Forked: yes  
Server Root: /usr/local/apache  
Config File: conf/httpd.conf
```

- **Configuración de un sólo módulo.** Utilice `http://server/server-info?module_name.c`. Por ejemplo, para ver información sobre el módulo `mod_cgi`, ejecute `http://server/server-info?mod_cgi.c`, que mostrará la siguiente información:

```
Module Name: mod_cgi.c  
Content handlers: (code broken)  
Configuration Phase Participation: Create Server Config,  
Merge Server Configs  
Module Directives:  
ScriptLog - the name of a log for script debugging info  
ScriptLogLength - the maximum length (in bytes) of the  
script debug log  
ScriptLogBuffer - the maximum size (in bytes) to record of a  
POST request  
Current Configuration:
```

- **Una lista con los módulos compilados.** Utilice `http://server/server-info?list`, que muestra la siguiente información:

```
mod_cgi.c  
mod_info.c  
mod_asis.c  
mod_autoindex.c  
mod_status.c  
prefork.c  
mod_setenvif.c  
mod_env.c
```

```
mod_alias.c  
mod_userdir.c  
mod_actions.c  
mod_imap.c  
mod_dir.c  
mod_negotiation.c  
mod_log_config.c  
mod_mime.c  
http_core.c  
mod_include.c  
mod_auth.c  
mod_access.c  
core.c
```

Por supuesto, su lista variará basándose en qué módulos tiene activados durante la configuración de la fuente. Ahora, vamos a ver cómo puede monitorizar el estado de un servidor Apache en ejecución.

## Permitir páginas de estado con `with mod_status`

El módulo `mod_status` permite a los administradores de Apache monitorizar el servidor mediante la Web. Se crea una página HTML con las estadísticas del servidor. Se genera, además, otra página de programación muy intuitiva. La información que se muestra en estas páginas incluye:

- El momento actual del sistema servidor
- El momento en el que se reinició el servidor por última vez
- El tiempo transcurrido desde que empezó a funcionar
- El número total de accesos al servidor hasta ese momento
- El número total de bytes transferidos hasta ese momento
- El número de solicitudes hijo servidas
- El número total de hijos desocupados
- El estado de cada hijo, el número de solicitudes que el hijo procesa y el número de bytes servidos por hijo
- Medias del número de solicitudes por segundo, el número de bytes servidos por segundo y el número de bytes por solicitud
- El porcentaje actual de CPU utilizada por cada hijo y el total utilizado por Apache
- El host y las solicitudes que se están procesando

Al igual que el módulo `mod_info`, este módulo tampoco está compilado por defecto en la distribución estándar de Apache, por lo que tiene que utilizar la

opción --enable-status en el script de configuración y compilar e instalar Apache.

**NOTA:** Parte de la información que acabamos de nombrar sólo está disponible cuando es capaz de desplegar ese tipo de información utilizando la directiva ExtendedStatus, que se discutirá más tarde en esta sección.

## Ver páginas de estado

Una vez que tiene el módulo mod\_status compilado y construido en su servidor Apache, necesita definir la localización URL que Apache debería utilizar para mostrar la información. En otras palabras, necesita decirle a Apache qué URL mostrará las estadísticas del servidor en su navegador Web.

Vamos a suponer que su nombre de dominio es domain.com, y que quiere utilizar la siguiente URL:

`http://www.domain.com/server-status`

Utilizando el contenedor <Location . . .>, puede decirle al servidor que quiere que maneje esta URL utilizando el manejador de estado del servidor que se encuentra en el módulo mod\_status. El siguiente contenedor es el encargado de llevar a cabo el trabajo:

```
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1 .domain.com
</Location>
```

A continuación, una vez que ha añadido la configuración en httpd.conf, reinicie el servidor y acceda a la URL desde un navegador. El contenedor <Location . . .> le permite acceder a la información de estado desde cualquier host de su dominio, o desde el servidor en sí. No olvide cambiar .domain.com a su verdadero nombre de dominio, ni olvide incluir el punto inicial.

**TRUCO:** Puede hacer que la página de estado se actualice a sí misma automáticamente utilizando la URL `http://server/server-status?refresh=N` para actualizar la página cada N segundos.

Para ver la información de estado extendida, añada la directiva ExtendedStatus On en el contexto de configuración del servidor. Por ejemplo, la

configuración completa relacionada con el estado del servidor en `httpd.conf` podría ser la siguiente:

```
ExtendedStatus On
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1 .domain.com
</Location>
```

A continuación se muestra un ejemplo de la información de estado ampliada:

Apache Server Status for rhat.nitec.com  
Server Version: Apache/2.0.14 (Unix)  
Server Built: Mar 14 2001 12:12:28

```
Current Time: Thursday, 15-Mar-2001 11:05:08 PST
Restart Time: Thursday, 15-Mar-2001 11:02:40 PST
Parent Server Generation: 0
Server uptime: 2 minutes 28 seconds
Total accesses: 17807 - Total Traffic: 529 kB
CPU Usage: u173.4 s.03 cu0 cs0 - 117% CPU load
120 requests/sec - 3660 B/second - 30 B/request
4 requests currently being processed, 8 idle servers
_WKKK.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
Scoreboard Key:  
" " Waiting for Connection, "S" Starting up, "R" Reading Request,  
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,  
"L" Logging, "G" Gracefully finishing, "." Open slot with no  
current process
```

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-0	0	0	0/87/87	-	0.07	1726072572	0	0.0	0.10	0.10	(unavailable)	
0-0	0	105	105/105	W	0.00	1726072572	0	50.5	0.05	0.05	(unavailable)	
0-0	0	166	166/166	K	0.02	1726072572	0	233.5	0.23	0.23	(unavailable)	
0-0	0	49	49/49	K	0.01	1726072572	0	25.2	0.02	0.02	(unavailable)	
0-0	0	77	77/77	K	0.08	1726072572	0	116.6	0.11	0.11	(unavailable)	

```
4-0 0 0/0/17323 _ 173.25 1726072572 0 0.0 0.00 0.00  
(unavailable)
```

```
-----  
Srv Child Server number - generation  
PID OS process ID  
Acc Number of accesses this connection / this child / this slot  
M Mode of operation  
CPU CPU usage, number of seconds  
SS Seconds since beginning of most recent request  
Req Milliseconds required to process most recent request  
Conn Kilobytes transferred this connection  
Child Megabytes transferred this child  
Slot Total megabytes transferred this slot  
-----
```

```
Apache/2.0.14 Server at rhat.nitec.com Port 80
```

## Simplificar el despliegue de estado

La página de estado desplegada por el módulo `mod_status` proporciona información extra que la hace inapropiada para utilizarla como archivo de datos para cualquier programa analizador de datos.

Así, por ejemplo, si quiere crear un gráfico para el estado de los datos de su servidor utilizando un programa de hojas de cálculo, tendría que borrar los datos manualmente. Sin embargo, el módulo proporciona un modo de crear salidas legibles para la máquina, para esa misma URL, modificándolas utilizando el `?auto` en `http://server/server-status?auto`. Un ejemplo de salida de estado sería el siguiente:

```
Total Accesses: 17855  
Total kBytes: 687  
CPULoad: 14.1982  
Uptime: 1221  
ReqPerSec: 14.6233  
BytesPerSec: 576.157  
BytesPerReq: 39.4001  
BusyServers: 8  
IdleServers: 8  
Scoreboard:  
_KKWKKKKK.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....
```

## Almacenar información del estado del servidor

Apache contiene un script Perl (lo puede encontrar en el directorio de soporte de la distribución fuente) llamado `log_server_status` que se puede utilizar para almacenar periódicamente la información del estado del servidor (utilizando la opción `auto`) en un archivo de texto.

Puede ejecutar este script como un trabajo del `cron` (demonio de Unix para la automatización de tareas) para aprovechar la información de estado, cada un cierto tiempo. Antes de que pueda utilizar el script, sin embargo, tiene que editar la fuente del script para modificar el valor de las variables `$wherelog`, `$port`, `$server` y `$request`. Los valores por defecto son:

```
$wherelog = "/var/log/graph/";
# Los registros serán del tipo "/var/log/graph/19960312"
$server = "localhost";
# Nombre del servidor, podría ser "www.foo.com"
$port = "80";
# Puerto del servidor
$request = "/status/?auto";
# Solicitud a enviar
```

En la mayor parte de los sitios funcionará lo siguiente:

```
$wherelog = "/var/log/apache";
$server = "localhost";
$port = "80";
$request = "/server-status?auto"
```

Podría necesitar realizar los siguientes cambios:

- Cambiar el valor de `$wherelog` con la ruta en la que le gustaría almacenar el archivo creado por el script. Asegúrese de que la ruta existe ya o tendrá que crear utilizando el nombre de ruta `mkdir -p`. Por ejemplo, `mkdir -p /var/log/apache` asegurará que todos los directorios (`/var`, `/var/log`, `/var/log/apache`) se van creando a medida que se necesitan.
- El valor de la variable `$port` debe ser el número de puerto del servidor que quiere monitorizar. El valor por defecto 80 es perfecto si su servidor se está ejecutando en un puerto HTTP estándar.
- La variable `$server` debería asignarse al nombre del host de su servidor. El valor por defecto del host local es adecuado si el script y el servidor se ejecutan en el mismo sistema. Si el servidor se encuentra en otra máquina es necesario marcar como valor el nombre completo del host (por ejemplo, `www.mydomain.com`).
- La variable `$request` debería fijarse con lo que se utilice en la directiva `<Location . . .>` más el script de consulta `?auto`.

Si no le gusta el formato de registro que utiliza el script, puede modificar la siguiente línea para ajustar sus necesidades:

```
print OUT "$time:$requests:$idle:$number:$cpu\n";
```

El script utiliza una conexión socket con el servidor Apache para enviar las solicitudes URL; por lo tanto, necesita asegurarse que tiene soporte socket para Perl. Por ejemplo, en un sistema Linux, el código Perl del socket se encuentra en `socket.ph`. Puede utilizar este `socket.ph` para determinar si existe este archivo en su sistema.

## Crear archivos de registro

Conocer la información del estado y de la configuración de su servidor es de gran ayuda a la hora de gestionar el servidor, pero conocer quién o qué está accediendo a su sitio o sitios Web es también muy importante, al tiempo que excitante. Puede obtener esta información utilizando las características de registro del servidor Apache. La siguiente sección discute cómo funciona el registro y cómo obtener el mejor beneficio de los módulos de registro de Apache.

Al tiempo que aparecieron los software de servidores Web en el mercado, fueron apareciendo muchos programas de análisis de registro en los servidores Web. Estos programas se convirtieron en parte del trabajo diario para muchos administradores Web. Todo esto conllevó a la era de las incompatibilidades de los archivos de registro, lo que hacía el análisis de registro difícil y engorroso; un solo programa de análisis no funcionaba en todos los archivos de registro. Entonces apareció la especificación *Common Log Format* (CLF). Esta especificación, permitía a todos los servidores Web, escribir registros de un modo razonablemente parecido, haciendo el análisis de registro entre servidores mucho más sencillo.

Por defecto, la distribución estándar de Apache incluye un módulo llamado `mod_log_config`, que es responsable del registro básico, y que escribe archivos de registro CLF por defecto. Puede cambiar su comportamiento utilizando la directiva `LogFormat`. Sin embargo, CLF cubre todas las necesidades en la mayoría de los entornos. A continuación se explica el contenido de cada línea de un archivo de registro CLF.

El archivo de registro CLF contiene una línea para cada solicitud. Una línea está compuesta por varias señales separadas por espacios:

```
host ident authuser date request status bytes
```

Si una señal no tiene un valor, entonces se representa por un guión (-). Las señales tienen los siguientes significados:

- `authuser`: si la URL solicitada requiere una autentificación HTTP Basic con éxito, entonces el nombre de usuario es el valor de esta señal.

- `bytes`: el número de bytes en el objeto devuelto al cliente, excluyendo todas las cabeceras HTTP.
- `date`: la fecha y hora de la última solicitud.
- `host`: el nombre completo de la máquina, dominio incluido, *fully qualified domain name*, del cliente, o su dirección IP.
- `ident`: si la directiva `IdentityCheck` está activada y la máquina del cliente ejecuta `identd`, entonces esta es la información que suministra el cliente.
- `request`: la línea de solicitud del cliente, encerrada entre dobles comillas (").
- `status`: el código de estado HTTP de tres dígitos que se devuelve al cliente.

Ver los apéndices para obtener una lista de todos los códigos de estado HTTP/1.1.

El campo de la fecha puede tener este formato:

```
date = [day/month/year:hour:minute:second zone]
```

El tamaño del campo de la fecha viene dado en la tabla 8.1.

**Tabla 8.1.** Tamaños de los campos

Campos	Valores
Día	2 dígitos.
Mes	3 letras.
Año	4 dígitos.
Hora	2 dígitos.
Minutos	2 dígitos.
Segundos	2 dígitos.
Zona	('+'   '-') 4*dígitos.

La siguiente sección ofrece un resumen de todas las directivas que puede utilizar con `mod_log_config`. Hay cuatro directivas disponibles en este módulo.

## Directiva TransferLog

`TransferLog` asigna el nombre del archivo de registro o programa al que se envía la información de registro. Por defecto, la información de registro se en-

cuenta en el formato CFL. Este formato se puede personalizar utilizando la directiva LogFormat. Observe que cuando la directiva TransferLog se encuentra dentro de un contenedor de host virtual, la información de registro se formatea utilizando la última directiva LogFormat que se encuentre dentro del contexto. Si la directiva LogFormat no se encuentra dentro del mismo contexto, por el contrario, se utiliza el formato de registro del servidor.

**Sintaxis:** TransferLog filename | " | path\_to\_external/program"

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

La directiva TransferLog toma como argumento o bien una ruta de un archivo de registro, o bien una tubería a un programa externo. Se considera que el nombre del archivo de registro es relativo al ServerRoot asignado si no se encuentra un carácter / inicial.

Por ejemplo, si ServerRoot está asignado a /etc/httpd, entonces la siguiente línea le dice a Apache que envíe la información de registro al archivo /etc/httpd/logs/access.log:

```
TransferLog logs/access.log
```

Cuando el argumento es una tubería a un programa externo. La información de registro se envía a la entrada estándar (STDIN) del programa externo.

**AVISO:** No se recomienda utilizar TransferLog con un programa externo ya que este hereda el TransferLog del servidor principal. Si se utiliza un programa, entonces se ejecuta bajo el usuario que inicio el httpd. Esto será la raíz si se inició el servidor como tal. Asegúrate de que el programa es seguro.

## Directiva LogFormat

LogFormat determina el formato del archivo de registro nombrado en la directiva TransferLog. Si incluye un nickname para el formato en la línea de la directiva, puede utilizarlo en otras directivas LogFormat y CustomLog en vez de repetir la cadena completa del formato. Una directiva LogFormat que define un nickname no hace nada más; es decir, solo define el nickname, y no aplica realmente el formato.

**Sintaxis:** LogFormat format [nickname]

**Predefinido:** LogFormat "%h %l %u %t \"%r\" %>s %b"

**Contexto:** Configuración del servidor, host virtual

Ver la sección "Personalizar sus archivos de registro" en este mismo capítulo, para obtener los detalles de las opciones de formato disponibles.

## Directiva CustomLog

Al igual que la directiva TransferLog, esta directiva le permite enviar información de registro a un archivo de registro o a un programa externo. A diferencia de TransferLog, sin embargo, le permite utilizar un formato de registro personalizado que se puede fijar como un argumento.

**Sintaxis:** CustomLog file | pipe [format | nickname]  
[env=[!]environment\_variable]

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

En el siguiente ejemplo, cada línea del archivo access.log se escribirá utilizando el formato los especificadores de formato dados. El formato determina un formato para cada línea del archivo de registro:

```
CustomLog logs/access.log "%h %l %u %t \"%r\" %>s %b"
```

Las opciones disponibles para el formato son exactamente las mismas que las que le sirven de argumento a la directiva LogFormat. Si el formato incluye cualquier espacio (lo cual ocurre en la mayoría de los casos), debe ir entre comillas dobles. En lugar de la cadena de formato real, puede utilizar un nickname definido con la directiva LogFormat. Por ejemplo:

```
LogFormat "%h %t \"%r\" %>s" myrecfmt  
CustomLog logs/access.log myrecfmt
```

Aquí access.log tendrá líneas en el formato myrecfmt.

**NOTA:** Las directivas TransferLog y CustomLog se pueden utilizar juntas para enviar datos en cada solicitud para conseguir que cada solicitud se registre en varios archivos de registro.

Por ejemplo, configurando la directiva TransferLog de la siguiente manera:

TransferLog logs/access.log myrecfmt

se enviará cada solicitud a los dos tipos de log. Esto significa que si el servidor envía una solicitud al cliente para cada solicitud, el resultado es que el cliente verá dos diferentes formatos de log.

Para terminar, si utiliza el mod\_setenvif (instalado por defecto) o el módulo para rescribir URL (mod\_rewrite, que no está instalado por defecto)

para determinar las variables de entorno basadas en una URL solicitada, puede crear registros condicionales utilizando la opción `env=[ ! ] environment variable` con la directiva `CustomLog`. Por ejemplo, imagine que permite a la gente bajarse un papel blanco PDF y quiere registrar todas las bajadas en un archivo de registro llamado `whitepaper.log` en su directorio de registro habitual. A continuación tenemos la configuración necesaria:

```
SetEnvIf Request_URI \.pdf$ whitepaper  
CustomLog logs/whitepaper.log common env=whitepaper  
CustomLog logs/access.log common env!=whitepaper
```

La primera línea determina la variable de entorno `whitepaper` cada vez que una URL solicitada tiene la extensión `.pdf`. Entonces, cuando se va a registrar la entrada, Apache utiliza los ajustes `env=whitepaper` en la primera directiva `CommonLog` para determinar si está asignada. Si lo está, se crea una entrada de registro en el archivo `logs/whitepaper.log` utilizando el formato habitual. Cuando no está asignada la variable de entorno `whitepaper`, la entrada de registro se crea utilizando el archivo `logs/access.log` de la forma habitual.

## Directiva `CookieLog`

`CookieLog` le permite registrar información sobre cookies en un archivo relacionado con la ruta de la directiva `ServerRoot`. No se recomienda esta directiva, porque no parece que Apache vaya a traer soporte para ella durante mucho tiempo. Para registrar datos sobre cookies, utilice el módulo de seguimiento de usuarios (`mod_usertrack`). El módulo de seguimiento de usuarios se discute más tarde en este capítulo.

**Sintaxis:** `CookieLog filename`

**Predefinido:** configuración del servidor, host virtual

## Personalizar sus archivos de registro

Aunque el formato por defecto, CLF, reúne la mayor parte de los requisitos de registro, a veces es útil ser capaz de personalizar o adaptar los datos de registro. Por ejemplo, podría desear registrar el tipo de navegadores que acceden a su sitio, para que su equipo de diseño Web pueda determinar el tipo de HTML específico de navegador a utilizar o evitar. O, quizás quiera saber qué sitios Web están enviando (es decir, remitiendo) visitantes a sus sitios. Todo esto es muy sencillo con Apache. El módulo de registro por defecto, `mod_log_config`, soporta registros personalizados. Los formatos personalizados están asignados con las directivas `LogFormat` y `CustomLog` del módulo. `LogFormat` y `CustomLog`

tienen una cadena como argumento. Esta cadena puede tener tanto caracteres literales como especificadores de formato especiales %. Cuando se utilizan valores literales en esta cadena, se copian en un archivo de registro para cada solicitud. Los especificadores %, sin embargo, son reemplazados con los valores correspondientes. Los especificadores % especiales se muestran en la tabla 8.2.

**Tabla 8.2.** Especificadores % especiales para entradas de registro

Especificador %	Descripción
%a	Dirección IP del cliente.
%A	Dirección IP del servidor.
%B	Bytes enviados, excluyendo las cabeceras HTTP; 0 para un byte no enviado.
%b	Bytes enviados, excluyendo las cabeceras HTTP; para un byte no enviado.
%c	Estado de conexión cuando la respuesta tiene lugar. El carácter "X" se escribe si la conexión fue abortada por el cliente antes de que se completase la respuesta. Si un cliente utiliza el protocolo keep-alive, se escribe un "+" para mostrar que la conexión se ha mantenido después de agotar el tiempo de operación. Se escribe un "-" para indicar que se cerró la conexión tras la respuesta.
%{mycookie}C	El contenido de una cookie llamada mycookie.
%D	La cantidad de tiempo (en microsegundos) que tarda en completar la respuesta.
%{myenv}e	El contenido de una variable de entorno llamada myenv.
%f	El nombre de archivo de la solicitud.
%h	El host remoto que realiza la solicitud.
%H	El protocolo de la solicitud (por ejemplo, HTTP 1/1).
%{ IncomingHeader }i	El contenido de IncomingHeader; es decir, la línea o líneas de cabecera en la solicitud enviada al servidor. El carácter i (incoming) al final, se refiere a que se trata de la cabecera de un cliente.
%l	Si la directiva IdentityCheck está activada y la máquina del cliente ejecuta identd, entonces esta es la información de identidad que suministra el cliente.

Especificador %	Descripción
%m	El método de la solicitud (GET, POST, PUT, etc.).
%{ ModuleNote }n	El contenido de <code>ModuleNote</code> desde otro módulo.
%{ OutgoingHeader }o	El contenido de <code>OutgoingHeader</code> ; es decir, la línea o líneas de cabecera en la respuesta. El carácter o ( <code>outgoing</code> ) al final quiere decir que se trata de una cabecera de un servidor.
%p	El puerto al que fue servida la solicitud.
%P	El ID del proceso del hijo que sirvió la solicitud.
%q	La cadena de consulta.
%r	La primera línea de la solicitud.
%s	El estado devuelto por el servidor en respuesta a la solicitud. Observe que cuando se redirige una solicitud, el valor de este especificador de formato continúa siendo el estado original de la solicitud. Si quiere almacenar el estado de la solicitud redirigida, utilice <code>%&gt;s</code> .
%t	Tiempo de la solicitud. El formato de tiempo es el mismo que en el formato CLF.
%{format}t	El tiempo, en la forma dada por el formato. (Puede ver además la página man de <code>strftime</code> en los sistemas Unix .)
%T	El tiempo que tarda el servidor en contestar, en segundos.
%u	Si la URL solicitada, necesita una autentificación HTTP Basic con éxito, entonces el nombre de usuario es el valor de este especificador de formato. El valor debería ser falso si el servidor devuelve un estado 401 (Necesita autentificación) después del intento de autentificación.
%U	La ruta de la URL solicitada.
%v	El nombre del servidor o del host virtual de la solicitud.
%V	El nombre del servidor para cada directiva <code>UseCanonicalName</code> .

Es posible incluir información condicional en cada uno de los especificadores que hemos visto. Las condiciones pueden ser la presencia (o ausencia) de ciertos

códigos de estados HTTP. Por ejemplo, imagine que quiere registrar todas las URL que dirigen a un usuario a una página que no existe. En ese caso, el servidor produce una cabecera de estado 404 (No encuentra). Por lo tanto, para registrar estas URL puede utilizar el especificador de formato:

```
'%404{Referer}i'
```

De igual modo, para registrar las URL resultantes de un estado inusual, puede utilizar:

```
'%!200,304,302{Referer}i'
```

Observe la utilización del carácter ! para indicar la ausencia de la lista de los estados del servidor. De igual modo, para incluir información adicional al final del especificador de formato CLF, puede extender el formato CLF, que es definido por la cadena de formato:

```
"%h %l %u %t \"%r\" %s %b"
```

Por ejemplo:

```
"%h %l %u %t \"%r\" %s %b \"%{Referer}i\" \"%{User-agent}i\"".
```

Estos registros de especificación CLF, formatean datos y añaden la información Referer y User-agent encontrada en las cabeceras proporcionadas por el cliente en cada entrada de registro. Hemos visto cómo añadir campos personalizados para el archivo de registro pero no sabemos qué es lo que ocurre si necesita almacenar estos datos en más de un archivo de registro. La siguiente sección discute cómo utilizar varios archivos de registro.

## Crear varios archivos de registro

En ocasiones, es necesario crear varios archivos de registro. Por ejemplo, si está utilizando un programa de análisis de registro que no puede manejar datos que no sean CLF, tendrá que escribir los datos que no son CLF en un archivo distinto. Puede crear varios archivos de registro fácilmente utilizando la directiva TransferLog y/o la directiva CustomLog del módulo mod\_log\_config. Simplemente repita estas directivas para crear más de un archivo de registro.

Si, por ejemplo, quiere crear un archivo de acceso CLF estándar y un archivo personalizado de todas las URL, puede utilizar algo parecido a esto:

```
TransferLog logs/access_log
CustomLog logs/referrer_log      "%{Referer}i"
```

Cuando tiene definidas TransferLog o CustomLog en la configuración del servidor principal, y tiene un host virtual definido, el registro relacionado con el host también se lleva a cabo en estos registros. Por ejemplo:

```
TransferLog logs/access_log
CustomLog logs/agents_log      "%{User-agent}i"

<Virtual Host 206.171.50.51>

    ServerName reboot.nitec.com
    DocumentRoot "/www/reboot/public/htdocs"
    ScriptAlias /cgi-bin/ "/www/reboot/public/cgi-bin/"

</VirtualHost>
```

El host virtual `reboot.nitec.com` no tiene definida una directiva `TransferLog` o una directiva `CustomLog` dentro de las etiquetas del host virtual. Toda la información de registro se almacenará en `logs/access_log` y en `logs/agents_log`. Si se añade la línea siguiente dentro del contenedor del host virtual:

```
TransferLog vhost_logs/reboot_access_log
```

Todos los registros del host virtual `reboot.nitec.com` se realizan en el archivo `vhost_logs/reboot_access_log`. No se utilizarán los archivos `logs/access_log` y `logs/agents_log` con el host virtual `reboot.nitec.com`.

## Registrar cookies

Hasta ahora, las opciones de registro discutidas, no le permiten identificar visitantes. La identificación de visitantes es muy importante, porque si sabe quién está realizando qué solicitudes, se puede hacer una idea de cómo se utiliza su contenido. Por ejemplo, imagine que tiene una página realmente buena es algún sitio de su página Web, y tiene algún modo de identificar a los visitantes en su registro. Si observa el registro y puede ver que muchos visitantes van de una página a otra hasta encontrar dicha página, podría reconsiderar su diseño Web y dejar la página más accesible. Apache tiene un módulo llamado `mod_usertrack` que le permite hacer un seguimiento de los visitantes a su sitio Web registrando las cookies HTTP.

**COOKIES HTTP:** Una cookie HTTP es simplemente una pieza de información que el servidor le pasa al navegador Web. Esta información se almacena normalmente en un par clave = valor y puede ir asociada con todo el sitio Web o con una URL particular en el sitio Web. Una vez que el servidor envía una cookie y el navegador la acepta, la cookie residirá en el sistema del navegador Web. Cada vez que el navegador Web solicite la misma URL, o cualquier URL que se encuentre bajo el alcance de la URL

**de la cookie, la información de la cookie se devuelve al servidor. Cuando asignamos la cookie, el servidor puede decirle al navegador Web que expire la cookie tras un tiempo determinado. El tiempo se puede determinar de modo que la cookie nunca se utilice en una sesión posterior, o se puede utilizar durante un largo periodo de tiempo.**

**Ha habido una gran controversia con la utilización de las cookies. Mucha gente considera las cookies como una intrusión de la privacidad. Utilizar cookies para hacer el seguimiento del comportamiento de usuarios es muy popular. De hecho, muchas compañías de anuncios de Internet hacen un gran uso de cookies para llevar a cabo el seguimiento de usuarios.**

Los datos de las cookies se escriben normalmente en un archivo de texto en un directorio de su software del navegador. Por ejemplo, utilizando la directiva CustomLog en el módulo de registro estándar, puede almacenar las cookies en archivos separados:

```
CustomLog logs/clickstream "%{cookie}C %r %t"
```

A continuación, vamos a ver el nuevo módulo mod\_usertrack.

Recuerde que mod\_usertrack no guarda un registro de cookies; simplemente genera cookies únicas para cada visitante. Puede utilizar CustomLog (como se vio antes) para almacenar estas cookies en un archivo de registro para su análisis.

La directiva mod\_usertrack no está compilada en la versión estándar de la distribución de Apache, por lo que necesita compilarla utilizando la opción --enable-usertrack antes de utilizarla. El módulo proporciona las directivas de las secciones siguientes.

## Directiva CookieExpires

Esta directiva se utiliza para determinar el período de expiración de las cookies que son generadas por este módulo. El período de expiración se puede definir en términos de número de segundos, o en un formato del tipo "1 month 2 days 3 hours."

**Sintaxis:** CookieExpires expiry-period

**Contexto:** configuración del servidor, host virtual

En el ejemplo siguiente, la primera directiva define el tiempo de expiración en segundos, y la segunda directiva define el tiempo de expiración utilizando el formato especial.

Observe que cuando el tiempo de expiración no está definido en un formato numérico, se asume el formato especial. Sin embargo, el formato especial requiere que coloque dobles comillas alrededor de la cadena de formato. Si no se utiliza esta directiva, las cookies sólo aguantan esa sesión del navegador.

```
CookieExpires 3600  
CookieExpires "2 days 3 hours"
```

## Directiva CookieTracking

Esta directiva activa o desactiva la generación automática de cookies. Cuando tiene asignado el valor `on`, Apache comienza a enviar cookies de seguimiento de usuarios para cada nueva solicitud. Esta directiva se puede utilizar para activar o desactivar este comportamiento en servidores o en directorios. Por defecto, compilar `mod_usertrack` no activa las cookies.

**Sintaxis:** `CookieTracking On | Off`

**Contexto:** configuración del servidor, host virtual, directorio, archivo control de acceso en el ámbito de directorio (`.htaccess`)

**Invalidar:** `FileInfo`

## Utilizar registros de error

Este capítulo ha discutido varias formas de registrar datos interesantes y fases de respuesta en cada transacción Web. Cuantos más datos obtenga sobre sus visitantes, más contento estará su departamento de marketing. Como administrador del sistema, sin embargo, estará feliz si todo marcha sin problemas. Apache le permite saber qué es lo que no funciona escribiendo registros de error. Sin registros de error, no será capaz de determinar qué es lo que va mal y dónde está teniendo lugar el error.

No es extraño que ese registro de error esté soportado por Apache en lugar de en un módulo del tipo `mod_log_config`.

La directiva `ErrorLog` le permite registrar todos los errores que encuentra Apache. Esta sección explora el modo en el que puede incorporar su registro de errores Apache en la facilidad `syslog` que se encuentra en la mayor parte de las plataformas Unix.

`Syslog` es el modo tradicional de registrar mensajes enviados por los procesos demonio (servidor). Podría preguntarse que si Apache es un demonio, por qué no puede escribir en `syslog`. De hecho, puede hacerlo. Todo lo que necesita hacer es reemplazar su directiva `ErrorLog` en el archivo de configuración por:

```
ErrorLog syslog
```

y reiniciar Apache. Utilizando un navegador Web, acceda a una página que no exista y compruebe si el archivo de registro `syslog` muestra una entrada `httpd`. Debería mirar su archivo `/etc/syslog.conf` para encontrar pistas sobre dónde aparecerán los mensajes `httpd`.

Por ejemplo, el listado 8.1 muestra `/etc/syslog.conf` para un sistema Linux.

#### Listado 8.1. /etc/syslog.conf

```
# Registra todos los mensajes del kernel para la consola.  
# Registra toda la pantalla.  
#kern.*                                /dev/console  
  
# Registra todo (excepto correo) del nivel de información o  
superior.  
# No registra mensajes privados de autentificación  
*.info;mail.none;authpriv.none          /var/log/messages  
  
# El archivo authpriv tiene acceso restringido.  
authpriv.*                               /var/log/secure  
  
# Registra todos los mensajes de correo en un solo sitio.  
mail.*                                    /var/log/maillog  
  
# Todo el mundo obtiene mensajes de emergencia,  
# los registra en otra máquina.  
*.emerg                                     *  
  
# Mantiene los errores en correo y en noticias en un nivel err  
# y superior, en un archivo especial.  
uucp,news.crit                            /var/log/spooler  
  
# Mantiene mensajes boot en boot.log  
local7.*                                   /var/log/boot.log
```

Hay dos líneas importantes (relacionadas con Apache) en el listado, que he marcado en negrita.

La primera línea (que empieza con **\*.info;mail.none;**) le dice a syslog que escriba todos los mensajes del tipo informativo (excepto para autentificación de correo y privada) en el archivo /var/log/messages, y la segunda línea (que empieza con **\*.emerg**) determina que todos los mensajes de emergencia deben escribirse en todos los archivos de registro. Utilizando la directiva LogLevel, puede especificar qué tipo de mensajes Apache deberían enviarse a syslog. Por ejemplo:

```
ErrorLog syslog  
LogLevel debug
```

Le está diciendo a Apache que envíe mensajes de depuración de errores a syslog. Si quiere almacenar mensajes de depuración de errores en un archivo distinto mediante syslog, entonces tiene que modificar /etc/syslog.conf. Por ejemplo:

```
*.debug                                /var/log/debug
```

Apache le permitirá guardar todos los mensajes de depuración en el archivo /var/log/debug si añade esta línea en /etc/syslog.conf y reinicia

`syslogd(kill -HUP syslogd_PID)`. Hay varios niveles de registro asignados:

- **Alert:** mensajes de alerta
- **Crit:** mensajes críticos
- **Debug:** mensajes registrados al nivel de depuración de errores que incluirán el archivo fuente y el número de línea en el que se genera el mensaje, para ayudar a la depuración de errores y al desarrollo del código
- **Emerg:** mensajes de emergencia
- **Error:** mensajes de errores
- **Info:** mensajes informativos
- **Notice:** mensajes de notificación
- **Warn:** advertencias

**TRUCO:** Si quiere ver las actualizaciones en `syslog` o en cualquier otro archivo de registro mientras que están ocurriendo, puede utilizar la utilidad `tail` que se encuentra en la mayoría de los sistemas Unix. Por ejemplo, si quiere ver las actualizaciones de un registro llamado `/var/log/messages` mientras que está ocurriendo, utilice:

```
tail -f /var/log/messages
```

## Analizar sus archivos de registro

Hasta ahora, ha aprendido a crear registros estándar basados en CLF y registros personalizados. Ahora, necesita un modo de analizar estos registros para utilizar los datos registrados. Sus necesidades de análisis pueden variar. En ocasiones necesitará producir largos informes, o quizás simplemente quiera verificar los registros. Para tareas sencillas, es mejor utilizar cualquiera que tenga a mano. La mayor parte de los sistemas Unix tienen utilidades suficientes y herramientas de script disponibles para llevar a cabo este trabajo.

Utilizando las utilidades Unix, puede recoger rápidamente la información necesaria; sin embargo, este método requiere algunos conocimientos de Unix, y no siempre es conveniente porque su jefe podría querer un informe "bonito" en lugar de una simple lista de texto. En tal caso, puede desarrollar su propio programa de análisis o utilizar una herramienta de análisis de terceras partes.

Vamos a emplear una utilidad Unix para obtener una lista de todos los hosts. Si usa la utilidad de registro por defecto o un registro personalizado con soporte CLF, puede encontrar fácilmente una lista de todos los host. Por ejemplo:

```
cat /path/to/httpd/access_log | awk '{print $1}'
```

imprime todas las direcciones IP (si tiene activada una búsqueda DNS [domain name server], cuando se muestran los alias de los host). La utilidad `cat` realiza una lista del archivo `access_log`, y el resultado se dirige al interpretador `awk`, que imprime únicamente el primer campo en cada línea utilizando la sentencia `print`. Esta sentencia imprime todos los host; pero ¿qué es lo que ocurre si quiere excluir los host de su red? En ese caso, debería utilizar:

```
cat /path/to/httpd/access_log | awk '{print $1}' | egrep -v '^206.171.50'
```

donde 206.171.50 debería reemplazarse con su dirección de red. He supuesto que tiene una red de tipo C. Si tiene una red de tipo B, sólo tiene que utilizar los primeros dos octetos de su dirección IP. Esta versión le permite excluir sus propios host utilizando la utilidad `egrep`, a la que se le indica que muestre (vía `-v`) sólo los host que no comienzan con la dirección de red 206.171.50. Sin embargo, esto no acaba de resultar totalmente satisfactorio, porque hay muchas posibilidades de repetición. Por lo tanto, la versión final es:

```
cat /path/to/httpd/access_log | awk '{print $1}' | uniq | egrep -v '^206.171.50'
```

La utilidad `uniq` filtra las repeticiones y muestra solo una lista por cada host. Por supuesto, si quiere ver el número total de host que tienen acceso a su sitio Web, puede dirigir el resultado final a la utilidad `wc` con la opción `-l`:

```
cat /path/to/httpd/access_log | awk '{print $1}' | uniq | egrep -v '^206.171.50' | wc -l
```

Esto le da el número total de conteo de líneas (es decir, el número de accesos del host).

Hay disponibles muchas herramientas de análisis de registro de servidores Web de terceras partes. La mayoría de estas herramientas esperan que los archivos de registro se encuentren en el formato CLF, por lo que es necesario que se asegure de tener un formato CLF en sus registros. La tabla 8.3 contiene una lista de estas herramientas y dónde encontrarlas.

**Tabla 8.3.** Herramientas de análisis de terceras partes

Nombre del producto	URL del producto
WebTrends	<a href="http://www.webtrends.com/">www.webtrends.com/</a>
Wusage	<a href="http://www.boutell.com/wusage/">www.boutell.com/wusage/</a>
Wwwstat	<a href="http://www.ics.uci.edu/pub/websoft/wwwstat/">www.ics.uci.edu/pub/websoft/wwwstat/</a>
Analog	<a href="http://www.statslab.cam.ac.uk/~sret1/analog/">www.statslab.cam.ac.uk/~sret1/analog/</a>

Nombre del producto	URL del producto
http-analyze	<a href="http://www.netstore.de/Supply/http-analyze/">www.netstore.de/Supply/http-analyze/</a>
Pwebstats	<a href="http://www.unimelb.edu.au/pwebstats.html">www.unimelb.edu.au/pwebstats.html</a>
WebStat Explorer	<a href="http://www.webstat.com/">www.webstat.com/</a>
AccessWatch	<a href="http://netpresence.com/accesswatch/">http://netpresence.com/accesswatch/</a>

El mejor modo de aprender qué herramienta le va a funcionar es probarlas todas, o al menos visitar sus sitios Web para comparar sus características. Dos utilidades que encuentro muy prácticas son Wusage y wwwstat. Wusage es mi aplicación de análisis de registro favorita. Es muy configurable y produce informes gráficos de alta calidad utilizando la librería de gráficos GD de la compañía. Wusage se distribuye en formato binario. Las copias de evaluación de wusage se distribuyen gratuitamente para varias plataformas Unix y Windows.

wwwstat es uno de mis programas de análisis gratuitos preferidos. Está escrito en Perl, por lo tanto, tiene que tener Perl instalado en el sistema en el que quiera ejecutar esta aplicación. Puede leer los resúmenes de salidas wwwstat con gwstat para producir gráficos de alta calidad en las estadísticas resumidas.

Crear registros en Apache es sencillo y útil. Crear registros le permite aprender más sobre lo que está ocurriendo en su servidor Apache. Los registros le ayudan a detectar e identificar los problemas de su sitio Web, a determinar las mejores características de su sitio Web, y mucho más. Pero debe haber alguna trampa. Lo que ocurre es que los archivos de registro ocupan mucho espacio disponible en el disco, por lo que es necesario un mantenimiento muy regular.

## Mantenimiento de registros

Permitiendo el registro, será capaz de ahorrar mucho trabajo, pero los registros en sí, le van a añadir trabajo extra: ha de mantenerlos. En los sitios Apache con muchos dominios virtuales, los archivos de registro se pueden convertir en enormes en poco tiempo, lo que puede causar fácilmente una crisis en el disco. Cuando los archivos de registro se vuelven muy grandes, debe rotarlos.

Tiene dos opciones para rotar sus registros: puede utilizar la utilidad de Apache llamada `rotatelog`, o puede utilizar `logrotate`, una facilidad disponible en la mayoría de los sistemas Linux.

### Utilizar `rotatelog`

Apache tiene soporte para una utilidad llamada `rotatelog`. Puede utilizar este programa del siguiente modo:

```
TransferLog "| /path/to/rotatelogs logfile  
rotation_time_in_seconds>"
```

Por ejemplo, si quiere rotar el registro de acceso cada 86.400 segundos (es decir, 24 horas), utilice la línea siguiente:

```
TransferLog "| /path/to/rotatelogs /var/logs/httpd 86400"
```

Cada acceso diario a la información de registro se almacenará en un archivo llamado `/var/logs/httpd.nnnn`, donde `nnnn` representa un número grande.

## Utilizar logrotate

La utilidad `logrotate` rota, comprime y envía archivos de registro. Está diseñada para facilitar la administración del sistema de archivos de registro. Permite la rotación, compresión, eliminación y envío automáticos de los archivos de registro a diario, mensualmente, o basándose en el tamaño. Normalmente, `logrotate` se ejecuta como un trabajo diario del cron (demonio de Unix para la automatización de tareas). Lea las páginas de información de `logrotate` para aprender más sobre él.

Si su sistema soporta la utilidad `logrotate`, debería crear un script llamado `/etc/logrotate.d/apache` tal y como se muestra en el listado 8.2.

**Listado 8.2.** `/etc/logrotate.d/apache`

```
'# Observe que este script supone que:  
#  
# a. Tiene instalado Apache en /usr/local/apache  
# b. Su ruta de registro es /usr/local/apache/logs  
# c. Su registro de acceso se llama access_log (por defecto en Apache)  
# d. Su registro de error se llama error_log (por defecto)  
# e. el archivo PID, httpd.pid, para Apache, se almacena en el  
#    directorio log (por defecto en Apache)  
#  
# Si alguna de estas suposiciones son falsas, por favor cambie  
# la ruta o el nombre de archivo.  
#  
/usr/local/apache/logs/access_log {  
    missingok  
  
    compress  
    rotate 5  
    mail webmaster@yourdomain.com  
    errors webmaster@yourdomain.com  
    size=10240K  
  
    postrotate  
        /bin/kill -HUP `cat /usr/local/apache/logs/httpd.pid  
        2>/dev/null` 2> /dev/null || true
```

```

        endscript
    }

/usr/local/apache/logs/error_log {
    missingok

    compress
    rotate 5
    mail webmaster@yourdomain.com
    errors webmaster@yourdomain.com
    size=10240K

    postrotate
        /bin/kill -HUP `cat /usr/local/apache/logs/httpd.pid
2>/dev/null` 2> /dev/null || true
    endscript
}

```

Esta configuración determina que tanto los accesos como los archivos de registro de error de Apache sean rotados cada vez que crece por encima de 10MB (10,240K), y que todos los archivos antiguos de registro sean comprimidos y enviados a webmaster@yourdomain.com una vez que han pasado por cinco rotaciones, en lugar de ser eliminados. Cualquier error que tenga lugar durante el procesamiento de los archivos de registros se envía a root@yourdomain.com.

## Utilizar logresolve

Por razones de rendimiento, debería deshabilitar la búsqueda de nombre de usuarios utilizando la directiva `HostNameLookups` con el valor `off`. Esto significa que sus entradas de registro mostrarán direcciones IP en lugar de nombres de host para los clientes remotos. Cuando se analizan los registros, es de gran ayuda tener los nombres de host de modo que pueda determinar quién accede a qué lugar fácilmente. Por ejemplo, a continuación tenemos unos cuantos ejemplos de entradas de registro en el archivo `/usr/local/apache/logs/access_log`.

```

207.183.233.19 - - [15/Mar/2001:13:05:01 -0800] "GET /book/
images/back.gif HTTP/1.1" 304 0
207.183.233.20 - - [15/Mar/2001:14:45:02 -0800] "GET /book/
images/forward.gif HTTP/1.1" 304 0
207.183.233.21 - - [15/Mar/2001:15:30:03 -0800] "GET /book/
images/top.gif HTTP/1.1" 304 0

```

Si tiene activada `HostNameLookups`, Apache traducirá las direcciones IP del cliente 207.183.233.19, 207.183.233.20 y 207.183.233.21 a los nombres de host apropiados; y si deja `LogFormat`:

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

y utiliza el formato común en registro utilizando CustomLog logs/access\_log, las muestras serán las siguientes:

```
nano.nitec.com - - [15/Mar/2001:13:05:01 -0800] "GET /book/images/back.gif HTTP/1.1" 304 0  
rhat.nitec.com - - [15/Mar/2001:14:45:02 -0800] "GET /book/images/forward.gif HTTP/1.1" 304 0  
r2d2.nitec.com - - [15/Mar/2001:15:30:03 -0800] "GET /book/images/top.gif HTTP/1.1" 304 0
```

Como activar la búsqueda DNS da lugar a que el servidor Apache tarde más tiempo en completar la respuesta, se recomienda que la búsqueda de nombres de host se realice de forma separada utilizando la utilidad logresolve, que se puede encontrar en el directorio bin (/usr/local/apache/bin) de Apache. El script log\_resolver.sh que se muestra en el listado 8.3 puede ejecutar esta utilidad.

#### Listado 8.3. log\_resolver.sh

```
#!/bin/sh

#
# Asegúrese de que cambia los nombres de las rutas según
# su instalación Apache
#

# Fully qualified path name (FQPN) de la
# utilidad log-resolver
LOGRESOLVER=/usr/local/apache/bin/logresolve

# Archivo estático generado por la utilidad
STATFILE=/tmp/log_stats.txt

# Archivo de registro de Apache
LOGFILE=/usr/local/apache/logs/access_log

# Nuevo archivo de registro que tiene traducida la dirección IP
OUTFILE=/usr/local/apache/logs/access_log.resolved

# Ejecuta el comando
$LOGRESOLVER -s $STATFILE < $LOGFILE > $OUTFILE

exit 0;
```

Cuando se ejecuta este script desde la línea de comando o desde un trabajo cron, crea un archivo llamado /usr/local/apache/logs/access\_log.resolved, que tiene todas las direcciones IP traducidas a sus nombres de host respectivos. Además, el script genera un archivo estático llamado /tmp/log\_stats.txt que muestra la información sobre el uso del caché,

las direcciones IP traducidas, y otra información de la utilidad resolver. A continuación se muestra un ejemplo de este tipo de archivos estáticos:

```
logresolve Statistics:  
Entries: 3  
  With name    : 0  
  Resolves     : 3  
Cache hits      : 0  
Cache size      : 3  
Cache buckets   :      IP number * hostname  
  130  207.183.233.19 - nano.nitec.com  
  131  207.183.233.20 - rhat.nitec.com  
  132  207.183.233.21 - r2d2.nitec.com
```

Observe que la utilidad no puede usar el caché porque las tres direcciones IP que están traducidas (para las entradas de registro mostradas) son únicas. Sin embargo, si su archivo de registro tiene direcciones IP del mismo host, se utilizará el caché para traducirlas en lugar de realizar solicitudes DNS a ciegas.

Si piensa que puede utilizar este script, le recomiendo que lo ejecute como un trabajo del cron. Por ejemplo, en mi servidor Web Apache ejecutando Linux, simplemente añado el script a /etc/cron.daily para crear una versión traducida del registro diario.





# 9

# Reescribir las URL

---

## En este capítulo

1. Trabajamos con el motor de reescritura de URL de Apache.
2. Analizamos la distribución URL.
3. Manejamos contenido.
4. Restringimos el acceso.

Las URL llevan a los visitantes a su sitio Web. Como administrador Apache, necesita asegurar que todas las URL posibles de su sitio Web son funcionales. ¿Cómo se hace? Ha de mantener la monitorización de los registros de error para todas las solicitudes URL fallidas. Si observa que las solicitudes se están devolviendo con un código de estado 404 Not Found, es el momento de investigar estas URL. A menudo, cuando el autor de un documento HTML actualiza un sitio Web, olvida que volver a nombrar un directorio puede dar lugar a la pérdida de muchos visitantes debido a la pérdida de coherencia en la carpeta de favoritos de estos.

Como administrador, ¿cómo podría resolver este problema? Las buenas noticias son que hay un módulo llamado `mod_rewrite` que le permite resolver

estos problemas y crear, además, soluciones muy interesantes utilizando reglas de reescritura de URL. Este capítulo discute este módulo y proporciona ejemplos prácticos de reescritura de URL.

## El motor de reescritura de URL de Apache

Cuando Apache recibe una solicitud URL, la procesa sirviendo el archivo al cliente (el navegador Web). ¿Qué ocurre si quiere intervenir en este proceso de integración de una URL a un archivo distinto o incluso a una URL distinta? Aquí es donde el módulo `mod_rewrite` muestra su valor. Le proporciona un mecanismo flexible para reescribir la URL solicitada en una nueva utilizando las reglas personalizadas de reescritura de URL. Una regla de reescritura de URL tiene la forma siguiente:

```
regex_pattern_to_be_matched  
regex_substitution_pattern
```

Sin embargo, también es posible añadir condiciones a una regla, de modo que la sustitución sólo se aplique si se encuentra la condición. Apache puede manejar estas URL sustitutas como si fueran subsolicitudes internas, o puede devolverlas al navegador Web como si fueran redirecciones externas. La figura 9.1 muestra un ejemplo en el que aparece una solicitud y el resultado de una regla `mod_rewrite`.

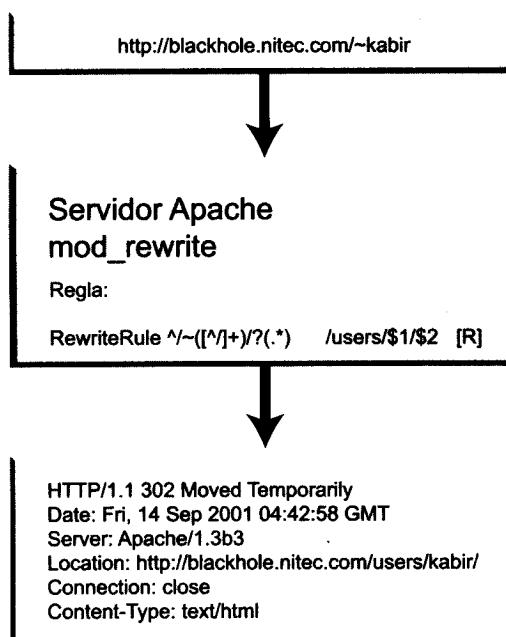


Figura 9.1. Ejemplo de una operación de reescritura de URL basada en reglas

La figura muestra una solicitud de `http://blackhole.nitec.com/~kabir` realizada desde el servidor Apache. El servidor recibe la solicitud y la pasa al módulo `mod_rewrite`, a la fase de traducción de la URL dentro del procesamiento de la solicitud. El módulo `mod_rewrite` aplica la regla de reescritura definida por una directiva llamada `RewriteRule`. En este ejemplo particular, la regla determina que si se encuentra un patrón del tipo `/~([^\n]+)/*(.*)`, debe ser reemplazado por `/users/$1/$2`. Como hay una indicación de redirección [R] en la regla, se enviará también al navegador Web una respuesta de redirección URL externa. La salida muestra como redirección a `http://blackhole.nitec.com/users/kabir/`.

Como puede ver, este tipo de redireccionamiento puede venir muy bien en muchas situaciones. Vamos a ver las directivas que le dan la opción de rescribir las URL. Debería familiarizarse también con las variables de servidor que se muestran en la tabla 9.1, que se pueden utilizar en muchas reglas y condiciones de reescritura.

**Tabla 9.1.** Variables del servidor disponibles para las reglas de reescritura de URL

Variable	Explicación
<code>SERVER_NAME</code>	Nombre del servidor en el host.
<code>SERVER_ADMIN</code>	Administrador de las direcciones de correo en el servidor Web.
<code>SERVER_PORT</code>	Dirección del puerto del servidor Web.
<code>SERVER_PROTOCOL</code>	Versión del protocolo HTTP que utiliza el servidor Web.
<code>SERVER_SOFTWARE</code>	Nombre del fabricante del servidor Web.
<code>SERVER_VERSION</code>	Versión del software del servidor Web.
<code>DOCUMENT_ROOT</code>	Directorio de documentos de máximo nivel en el sitio Web.
<code>HTTP_ACCEPT MIME</code>	Tipos aceptables para el cliente Web.
<code>HTTP_COOKIE</code>	Cookie recibida desde el cliente Web.
<code>HTTP_FORWARDED</code>	URL enviada.
<code>HTTP_HOST</code>	Nombre del host del servidor Web.
<code>HTTP_PROXY_CONNECTION</code>	La información de la conexión proxy HTTP.
<code>HTTP_REFERER</code>	La URL que se refiere a la URL actual.
<code>HTTP_USER_AGENT</code>	Información sobre el cliente Web.
<code>REMOTE_ADDR</code>	Dirección IP del cliente Web.

Variable	Explicación
REMOTE_HOST	Nombre del host del cliente Web.
REMOTE_USER	Nombre de usuario del usuario autenticado.
REMOTE_IDENT	Información sobre la identificación del usuario remoto.
REQUEST_METHOD	Método de solicitud HTTP utilizado para solicitar la URL actual.
SCRIPT_FILENAME	Ruta física del archivo de script solicitado.
PATH_INFO	Ruta de la URL solicitada.
QUERY_STRING	Datos de consulta enviados junto con la URL solicitada.
AUTH_TYPE	Tipo de autenticación utilizada.
REQUEST_URI	URI solicitado.
REQUEST_FILENAME	Igual que <code>SCRIPT_FILENAME</code> .
THE_REQUEST	URL solicitada.
TIME_YEAR	Año actual.
TIME_MON	Mes actual.
TIME_DAY	Día actual.
TIME_HOUR	Hora actual.
TIME_MIN	Minuto actual.
TIME_SEC	Segundo actual.
TIME_WDAY	Día de la semana actual.
TIME	Momento actual.
API_VERSION	Versión del API utilizado.
IS_SUBREQ	Determina si la solicitud es una subsolicitud.

## RewriteEngine

Esta directiva le proporciona un mecanismo de activación/desactivación para el motor de reescritura de URL en el módulo `mod_rewrite`. Por defecto, la reescritura está desactivada. Para utilizar el motor de reescritura, debe activar el motor asignándole a esta directiva el valor `on`.

**Sintaxis:** `RewriteEngine On | Off`

**Predefinido:** `RewriteEngine Off`

**Contexto:** configuración del servidor, host virtual, archivo de control de acceso en el ámbito de directorio (`.htaccess`)

Cuando se permite la reescritura de URL en la configuración del archivo de directorios (`.htaccess`), debe activar (es decir, darle el valor `On`) a esta directiva dentro de cada archivo de configuración de directorios y asegurarse de que está activando la siguiente directiva en el contexto adecuado para el directorio:

```
Options FollowSymLinks
```

En otras palabras, si el directorio pertenece a un sitio de un host virtual, asegúrese de que esta opción está activada dentro del contenedor de host adecuado. Del mismo modo, si el directorio en cuestión forma parte del espacio de documentos del servidor Web principal, asegúrese de que esta opción está activada en la configuración del servidor principal.

**NOTA:** Activar las reglas de reescritura a nivel de las configuraciones de directorio podría disminuir el rendimiento de su servidor Apache. Esto se debe a que `mod_rewrite` emplea un truco para soportar reglas de escritura a nivel de directorio, y este truco implica el aumento de la carga de procesos en el servidor. Por tanto, debe evitar utilizar reglas de reescritura en los archivos de configuración a nivel de directorio siempre que sea posible.

## RewriteOptions

Esta directiva le permite determinar opciones para cambiar el comportamiento del motor de reescritura. Actualmente, la única opción disponible es `inherit`. Dándole el valor `inherit` a esta directiva, puede forzar que una configuración de bajo nivel herede la configuración de mayor nivel.

**Sintaxis:** `RewriteOptions option1 option2 [...]`

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual, archivo de control de acceso en el ámbito de directorio (`.htaccess`)

Por ejemplo, si asigna esta directiva en el área de configuración de su servidor principal, un host virtual definido en el archivo de configuración heredará toda la configuración rescrita, así como las reglas de reescritura, las condiciones, las integraciones, y similares. De igual modo, cuando esta directiva se asigna en un archivo de configuración en el ámbito y las integraciones, por defecto, el motor de reescritura no permite heredar una configuración rescrita, pero esta directiva le permite alterar el comportamiento por defecto.

# RewriteRule

Esta directiva le permite definir una regla de reescritura. La regla debe tener dos argumentos. El primer argumento es el patrón buscado que debe cumplir para aplicar la cadena de sustitución. El patrón buscado se escribe utilizando una expresión regular (ver apéndice B para obtener los conceptos básicos sobre las expresiones regulares). La cadena de sustitución puede construirse con todo texto, con referencias a las subcadenas en el patrón de búsqueda, con valores para las variables del servidor, e incluso con funciones map. La lista de indicadores puede contener una o más cadenas de indicadores, separados por comas, para informar al motor de reescritura sobre qué es lo que hay que hacer después con la sustitución.

**Sintaxis:** RewriteRule search\_pattern substitution\_string [flag\_list]

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual, archivo de control de acceso en el ámbito de directorio (.htaccess)

Vamos a ver un ejemplo:

```
RewriteRule /~([/]+)/?(.*) /users/$1/$2 [R]
```

Aquí, el patrón de búsqueda es /~([/]+)/?(.\*) y la cadena de sustitución es /users/\$1/\$2. Observe la utilización de referencias en la cadena de sustitución. La primera cadena de referencia \$1 corresponde a la cadena encontrada en el primer conjunto de paréntesis (desde la izquierda). Por lo tanto \$1 está asignado a cualquiera que coincida con ([/]+) y \$2 con la siguiente cadena encontrada en (.\*) . Cuando una solicitud URL es como la siguiente:

```
http://blackhole.evoknow.com/~kabir/welcome.html
```

El valor de \$1 es kabir, y \$2 es welcome.html; por lo que la cadena de sustitución será la siguiente:

```
/users/kabir/welcome.html
```

Cuando tiene especificada más de una RewriteRule, la primera RewriteRule opera en la URL original y si tiene lugar una coincidencia, la segunda regla no vuelve a operar en la URL original. En lugar de eso, toma la URL sustituida por la primera regla como la URL en la que ha de aplicar las reglas.

En un escenario en el que hay una coincidencia a cada paso, una asignación de tres reglas de reescritura funcionará del siguiente modo:

```
RewriteRule search-pattern-for-original-URL substitution1  
[flags]
```

```

RewriteRule search-pattern-for-substitution1 substitution2
[flags]
RewriteRule search-pattern-for-substitution2 substitution3
[flags]

```

Es posible aplicar más de una regla a la URL original utilizando el indicador C para decirle al motor de reescritura cómo encadenar varias reglas. En este caso, no debería realizar una sustitución hasta que se hayan aplicado todas las reglas, de modo que pueda utilizar una cadena especial de sustitución para permitir una sustitución en una regla. La tabla 9.2 tiene una lista de los detalles de los indicadores posibles.

**Tabla 9.2.** Indicadores RewriteRule

Indicador	Significado
C   chain	Este indicador determina que la regla actual se encadene con la siguiente regla. Cuando una regla se encadena con un indicador C, sólo se utiliza si ha habido coincidencia con la regla anterior. Cada regla en la cadena da lugar a una coincidencia. Cada regla de la cadena debe contener el indicador, y si la primera regla no encuentra coincidencia, se ignora la cadena completa de reglas.
E=var:value   env=var:value	Puede determinar una variable de entorno utilizando esta directiva. La variable es accesible para las condiciones de reescritura, los Server Side Includes, los scripts CGI, y similares.
F   forbidden	Cuando una regla, que está utilizando este indicador, encuentra coincidencia, envía al navegador una cabecera de respuesta HTTP llamada FORBIDDEN (código de estado 403). Esto deshabilita con eficacia la URL solicitada.
G   gone	Cuando una regla, que está utilizando este indicador, encuentra coincidencia, envía al navegador una cabecera de respuesta HTTP llamada GONE (código de estado 410). Esto le dice al navegador que la URL solicitada no está disponible en este servidor.
L   last	Le dice al motor de reescritura que termine el procesamiento de reglas inmediatamente para que no se apliquen más reglas a la nueva URL tras la sustitución.

Indicador	Significado
N   next	Le dice al motor de búsqueda que vuelva a empezar desde la primera regla. Sin embargo, la primera regla no vuelve a intentar una coincidencia con la URL original, porque ahora opera sobre la URL nueva producto de la sustitución. Esto crea un loop. Debe tener condiciones de término en el loop para evitar un loop infinito.
NC   nocase	Le dice al motor de reescritura que no haga distinción entre mayúsculas y minúsculas en la búsqueda de coincidencias.
NS   nosubreq	Utilice este indicador para evitar aplicar una regla en una solicitud URL generada internamente.
P   proxy	El uso de este indicador convertirá una solicitud URL en una solicitud proxy interna. Esto sólo funcionará si tiene compilado Apache con el módulo <code>mod_proxy</code> y configurado para utilizar el módulo del proxy.
QSA   qsappend	Este indicador le permite adjuntar datos (del tipo pares clave = valor) a la parte de la cadena de la consulta de la URL producida tras la sustitución.
R [= HTTP code]   redirect	Fuerza redirecciones externas de los clientes mientras que prefija la sustitución con <code>http://server[:port]/</code> . Si no viene dado el código de respuesta HTTP, se utiliza el código de respuesta por defecto 302 (MOVED TEMPORARILY). Esta regla se debería utilizar con L o con el indicador last.
S=n   skip=n	Se salta las n reglas siguientes.
T=MIME-type   type=MIME-type	Fuerza a que el MIME-type especificado, sea el MIME-type del archivo objetivo de la solicitud.

**NOTA:** Puede añadir condiciones a sus reglas precediéndolas con una o varias directivas `RewriteCond`, que se discuten en la sección siguiente.

## RewriteCond

La directiva RewriteCond es útil cuando queremos añadir una condición extra para una regla de reescritura especificada por la directiva RewriteRule. Puede tener varias directivas RewriteCond para cada RewriteRule. Deben definirse todas las condiciones de reescritura antes de la regla en sí.

**Sintaxis:** RewriteCond test\_string condition\_pattern [flag\_list]

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual, archivo de control de acceso en el ámbito de directorio (.htaccess)

La cadena de prueba debe estar construida con texto simple, variables del servidor o referencias, tanto de la regla de reescritura actual como de la última condición de reescritura. Para acceder a la referencia nth de la última directiva RewriteRule, utilice \$n; para acceder a la referencia nth de la última directiva RewriteCond, utilice %n.

Para acceder a la variable del servidor, utilice el formato %{variable name}. Por ejemplo, para acceder a la variable REMOTE\_USER, especifique %{REMOTE\_USER} en la cadena de prueba.

La tabla 9.3 contiene una lista de varios formatos de acceso a datos.

**Tabla 9.3.** Formatos de acceso a datos para la directiva RewriteCond

Especificador de formato	Significado
%{ENV:variable}	Utilícelo para acceder a cualquier variable de entorno disponible en el proceso Apache.
%{HTTP:header}	Utilícelo para acceder a la cabecera HTTP utilizada en la solicitud.
%{LA-U:variable}	Utilícelo para acceder al valor de la variable que no está disponible en la fase actual del proceso. Por ejemplo, si necesita utilizar la variable del servidor REMOTE_USER en una condición de reescritura almacenada en el archivo de configuración del servidor (httpd.conf), no puede utilizar %{REMOTE_USER} porque esta variable sólo se define una vez que el servidor ha llevado a cabo la fase de autenticación, que tiene lugar después de la fase de procesamiento de URL del módulo mod_rewrite. Para saber qué usuario es el que da lugar a una autenticación exitosa del nombre de usuario,

Especificador de formato	Significado
	puede utilizar <code>%{LA-U:REMOTE_USER}</code> . Sin embargo, si está accediendo a datos <code>REMOTE_USER</code> de un <code>RewriteCond</code> en un archivo de configuración de directorios, puede utilizar <code>%{REMOTE_USER}</code> porque la fase de autorización ha terminado ya y la variable del servidor está disponible como siempre. La búsqueda tiene lugar generando una subsolicitud basada en URL.
<code>%{LA-F:variable}</code>	Igual que <code>%{LA-U:variable}</code> en la mayoría de los casos, pero la búsqueda se realiza utilizando una subsolicitud interna basada en el nombre de archivo.

El patrón de condición puede utilizar también alguna notación determinada además de ser una expresión regular. Por ejemplo, puede realizar comparaciones léxicas entre la cadena de prueba y el patrón prefijando el patrón con un `<`, un `>`, o un `=`. En este caso, el patrón se compara con la cadena de prueba como una cadena de solo texto.

Es posible que, en algunas ocasiones, quiera comprobar si la cadena de prueba es un archivo, un directorio o un enlace simbólico. En este caso, puede reemplazar el patrón con cadenas especiales que puede encontrar en la tabla 9.4.

**Tabla 9.4.** Opciones condicionales para la cadena de prueba en la directiva `RewriteCond`

Opciones condicionales	Significado
<code>-d</code>	Comprueba si existe el directorio especificado por la cadena de prueba.
<code>-f</code>	Comprueba si existe el archivo especificado por la cadena de prueba.
<code>-s</code>	Comprueba si existe el archivo de tamaño distinto de cero especificado por la cadena de prueba.
<code>-l</code>	Comprueba si existe el enlace simbólico especificado por la cadena de prueba.
<code>-F</code>	Comprueba si el archivo especificado por la cadena de prueba existe y es accesible.
<code>-U</code>	Comprueba si la URL especificada por la cadena de prueba es válida y accesible.

Puede utilizar ! delante de estas condiciones para negar sus significados. La lista de indicadores opcionales puede consistir en una o más cadenas separadas por comas tal y como se muestra en la tabla 9.5.

**Tabla 9.5.** Opciones de indicadores para la directiva RewriteCond

Indicador	Significado
NC   nocase	Realiza una prueba de condiciones sin hacer distinción entre mayúsculas y minúsculas.
OR   ornext	Normalmente, cuando tiene más de una RewriteCond para una directiva RewriteRule, estas condiciones se añaden juntas para que tenga lugar la sustitución final. Sin embargo, si necesita crear una relación OR entre dos condiciones, utilice este indicador.

## RewriteMap

La directiva RewriteMap facilita una búsqueda clave valor a través de la utilización de un mapa. Piense en este mapa como una tabla de datos en la que cada fila tiene una clave y un valor. Normalmente, un mapa está almacenado en un archivo. Puede ser un archivo de texto, un archivo DBM, una función interna de Apache, o un programa externo. El tipo de mapa se corresponde con la fuente del mapa. La tabla 9.6 contiene una lista de tipos aplicables de mapas.

**Sintaxis:** RewriteMap name\_of\_map type\_of\_map:source\_of\_map

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

**Tabla 9.6.** Opciones de indicadores para la directiva RewriteMap

Tipo de mapa	Descripción
txt	Archivo de texto que tiene líneas de valores de claves de modo que cada par clave valor se encuentra en una sola línea separados por, al menos, un carácter en blanco. El archivo puede contener líneas de comentarios que empiezan por # o puede tener líneas en blanco. Tanto los comentarios como las líneas en blanco son ignorados. Por ejemplo: Key1 value1

Tipo de mapa	Descripción
	Key2 value2 Define dos pares clave valor. Observe que los mapas basados en archivos de texto se leen durante el arranque de Apache y sólo se vuelven a leer si los archivos se han actualizado una vez que el servidor está preparado y ejecutándose. Los archivos también se vuelven a leer durante el reinicio del servidor.
rnd	Un tipo especial de archivo de texto, que tiene todas las restricciones del tipo <code>txt</code> pero que permite flexibilidad en la definición del valor. El valor de cada clave se puede definir como un conjunto de valores ORed utilizando el carácter   (barra vertical). Por ejemplo:  Key1 first_value_for_key1   second_value_for_key1 Key2 first_value_for_key2   second_value_for_key2 Define dos pares de clave valor en las que cada clave tiene varios valores. El valor seleccionado se decide de forma aleatoria.
Int	Las funciones internas de Apache <code>toupper(key)</code> y <code>tolower(key)</code> se pueden utilizar como fuentes de mapas. La primera función convierte todas las letras de la llave en letras mayúsculas y la segunda convierte todas las letras de la clave en letras minúsculas.
dbm	Se puede utilizar un archivo DBM como fuente de mapas. Esto puede ser muy útil y rápido (comparado con los archivos de texto) cuando tiene un gran número de pares clave valor. Tenga en cuenta que estos mapas basados en archivos DBM sólo se leen durante el arranque de Apache y únicamente se vuelven a leer si el archivo se actualiza una vez que el servidor está preparado y ejecutándose. Los archivos también se vuelven a leer durante el reinicio del servidor.
prg	Un programa externo puede generar el valor. Cuando se utiliza un programa externo, se inicia en el arranque de Apache y los datos (clave, valor) se transfieren entre Apache y el programa mediante una entrada estándar ( <code>stdin</code> ) y una salida estándar ( <code>stdout</code> ). Asegúrese de que utiliza la directiva <code>RewriteLock</code> para definir un archivo de bloqueo cuando utiliza un programa externo, así como de que lee la entrada desde <code>stdin</code> y la escribe en <code>stdout</code> en un modo I/O que no se encuentre en un buffer.

## RewriteBase

Esta directiva sólo es útil si está utilizando reglas de escritura en archivos de configuración a nivel de directorio. Sólo es necesario para las rutas de URL que no integran en el directorio físico del archivo objetivo. Asigne a esta directiva el alias que utilice para el directorio. Esto asegurará que mod\_rewrite utilizará el alias en lugar de la ruta física en la URL final (sustituida).

**Sintaxis:** RewriteBase base\_URL

**Predefinido:** ruta actual de directorio de la configuración para directorios (.htaccess)

**Contexto:** archivo de control de acceso a directorios (.htaccess)

Por ejemplo, cuando se asigna un alias del siguiente modo:

```
Alias /icons/ "/www/nitec/htdocs/icons/"
```

y las reglas de reescritura están disponibles en el archivo /www/nitec/htdocs/icons/.htaccess, la directiva RewriteBase debe determinarse del siguiente modo:

```
RewriteBase /icons/
```

## RewriteLog

Si quiere registrar las aplicaciones de sus reglas de reescritura, utilice esta directiva para asignar un nombre de archivo de registro. Al igual que el resto de las directivas, supone que una ruta sin barra inclinada inicial (/) significa que quiere escribir un archivo de registro en el directorio raíz del servidor.

**Sintaxis:** RewriteLog path\_to\_logfile

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

Por ejemplo, la siguiente directiva escribe un archivo de registro en el subdirectorío de registros bajo el directorio raíz de su servidor:

```
RewriteLog logs/rewrite.log
```

Tal y como se mencionó antes, únicamente el usuario del servidor puede escribir en un registro escrito por un servidor.

## RewriteLogLevel

Esta directiva le permite especificar qué se ha registrado en el archivo de registros. El valor por defecto 0 significa que no se va a registrar nada. De hecho,

un nivel de registro de 0 significa que no va a producirse ningún proceso relacionado con registros dentro de este módulo. Por lo tanto, si quiere deshabilitar el registro, mantenga este valor en 0.

**Sintaxis:** RewriteLogLevel level

**Predefinido:** RewriteLogLevel 0

**Contexto:** configuración del servidor, host virtual

**NOTA:** Si fija la directiva RewriteLog en /dev/null y RewriteLogLevel en un valor que no sea 0, se llevará a cabo el proceso interno relacionado con el registro, pero no se producirá ningún registro. Este proceso gasta gran cantidad de recursos de su sistema, por lo que si no quiere registro, mantenga el valor por defecto en esta directiva. Tiene 10 niveles de registro, que van de 0 a 9. Cuanto más alto sea el nivel, más datos de registro se escriben.

## RewriteLock

La directiva RewriteLock le permite determinar un programa de mapeado externo para crear mapas de reescritura. Cuando utilice la directiva RewriteLock tiene que especificar un nombre de registro. Este archivo se utiliza como archivo de bloqueo para comunicación sincronizada con los programas externos de mapeo.

**Sintaxis:** RewriteLock filename

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

## Distribución de las URL

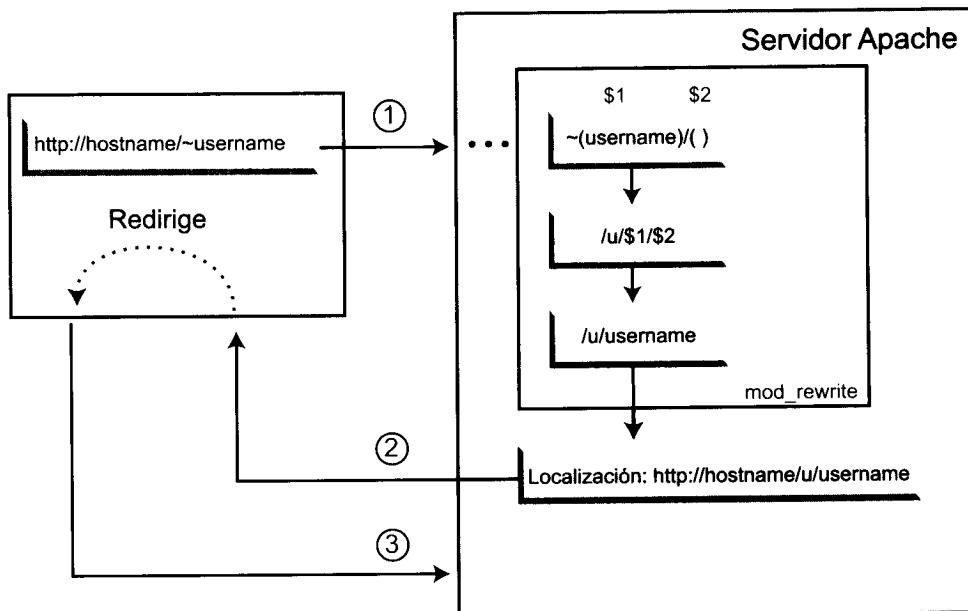
Esta sección proporciona ejemplos de URL reescritas que trabajan con la distribución de las URL. A menudo, necesitará redireccionar o ampliar una solicitud URL a otra URL. Los siguientes ejemplos le muestran cómo mod\_rewrite puede ayudar en estos casos.

### Ampliar una URL a la forma canónica de las URL

Los sitios Web que ofrecen páginas home para los usuarios, habitualmente soportan un esquema URL del siguiente tipo:

`http://hostname/~username`

Esta es una URL de atajo y necesita estar integrada a una URL canónica. Podría tener también otros atajos u otras URL internas que necesitasen estar integradas en sus URL canónicas. Este ejemplo le muestra cómo se traduce `~username` a `/u/username`. La figura 9.2 ilustra lo que tiene que ocurrir.



**Figura 9.2.** Ampliar una URL solicitada a una URL canónica

Cuando se recibe una solicitud para `http://hostname/~username` en (1), la regla de reescritura la traducirá en `/u/username` y redirigirá al navegador a la nueva URL (2). El navegador volverá a solicitar la URL `http://hostname/u/username` en (3) y el procesamiento habitual de solicitudes en Apache completará el proceso de la solicitud.

Es necesario redirigir el HTTP externo porque cualquier solicitud posterior debe utilizar también la URL canónica traducida en lugar de `~username`. La regla tiene que hacer lo siguiente:

```
RewriteRule ^/~([^\/]+) /?(.*) /u/$1/$2 [R,L]
```

**NOTA:** Observe que el indicador R se utiliza para redirigir, y el indicador L se utiliza para indicar que no se puede aplicar otra regla de reescritura a la URL sustituida.

Muchos sitios ISP con miles de usuarios utilizan una distribución estructurada de directorios home; es decir, cada directorio home es un subdirectorio que empieza, por ejemplo, con el primer carácter del nombre de usuario. Por lo tanto,

`~foo/anypath` es `/home/f/foo/www/anypath`, mientras que `~bar/anypath` es `/home/b/bar/www/anypath`. Para implementar un esquema de traducción de las URL a la forma canónica de las URL, en este caso, se puede utilizar la siguiente regla:

```
RewriteRule ^/~(([a-z])[a-zA-Z0-9]+)(.*) /home/$2/$1/www$3 [R,L]
```

## Redirigir un directorio home de usuario a un nuevo servidor Web

Si tiene muchas páginas home de usuarios en un servidor Web y necesita moverlas a una máquina nueva por alguna razón, necesita tener una regla de redirección parecida a la que se muestra en la figura 9.3.

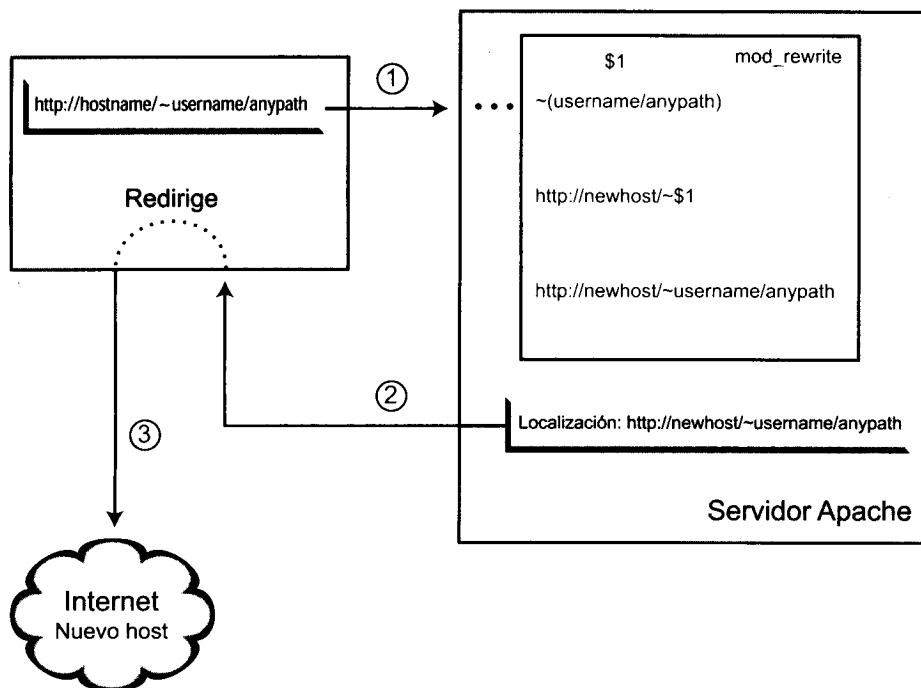


Figura 9.3. Redirigir directorios home de usuarios a un nuevo servidor Web

La solución es muy fácil con `mod_rewrite`. Cuando el navegador solicita `http://hostname/~username/anypath` como se muestra en (1) en la figura, el servidor Web la traduce a `http://newhost/~username/anypath` como se muestra en (2) y redirige el navegador a esta nueva localización. En el servidor Web antiguo (el que redirige la URL) simplemente redirige todas las URL `~/user/anypath` a `http://newhost/~user/anypath`, del siguiente modo:

```
RewriteRule ^/~(.+) http://newhost/~$1 [R,L]
```

# Buscar una página en varios directorios

En ocasiones es necesario dejar que el servidor Web busque páginas en más de un directorio. Ni MultiViews ni otras técnicas de este tipo nos pueden ayudar en esta ocasión. Por ejemplo, imagine que quiere manejar una solicitud para `http://hostname/filename.html` de modo que si `filename.html` no se encuentra en el directorio `dir1` de su servidor Web, el servidor lo intenta en el subdirectorio `dir2`. La figura 9.4 ilustra lo que tiene que ocurrir.

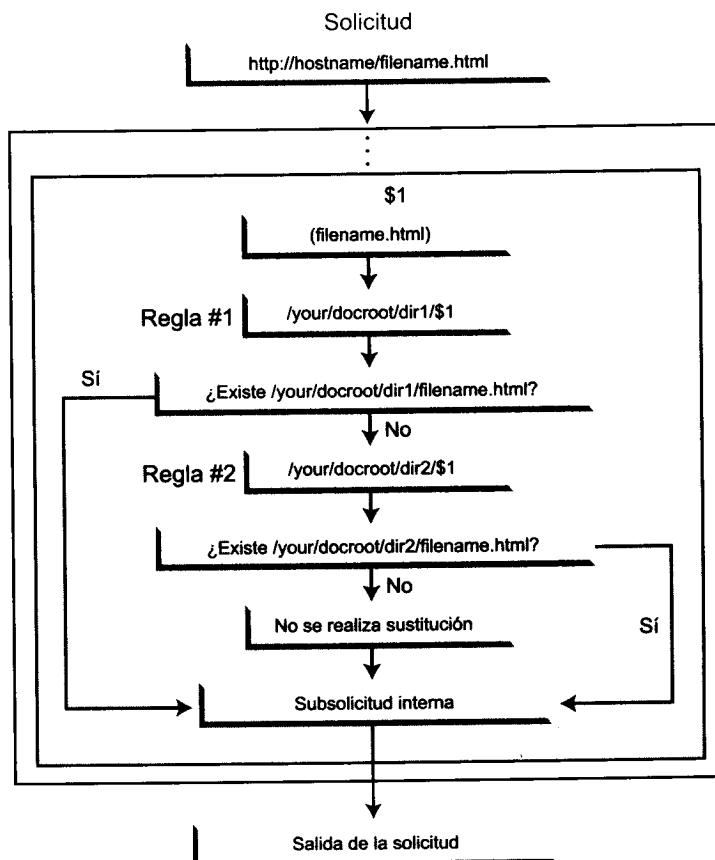


Figura 9.4. Buscar una página en varios directorios

Las reglas que hay que implementar son:

```
RewriteCond      /your/docroot/dir1 %{REQUEST_FILENAME} -f
RewriteRule ^.+  /your/docroot/dir1$1 [L]

RewriteCond      /your/docroot/dir2 %{REQUEST_FILENAME} -f
RewriteRule ^.+  /your/docroot/dir2$1 [L]

RewriteRule ^.+ - [PT]
```

La primera regla sustituye la URL solicitada por /your/docroot/dir1/\$1 (donde \$1 es el archivo objetivo en la solicitud), sólo si el archivo solicitado existe en el subdirectorío your/docroot/dir1/. Si se satisface la condición, esta es la última regla aplicada a esta URL. Sin embargo, si no se encuentra una coincidencia, se aplica la siguiente regla. Esta regla hace lo mismo que la primera pero utiliza el subdirectorío dir2 para la ruta. Esta regla se convierte en la última en el caso de encontrar una coincidencia. En el caso de que no se satisfaga ninguna de las reglas, la solicitud no se sustituye y pasa al procesamiento normal.

Para comprobar que la regla utiliza el registro, primero, desactive el registro de reescritura añadiendo las siguientes directivas antes de las reglas anteriores.

```
RewriteLog logs/rewrite.log  
RewriteLogLevel 5
```

El registro de reescritura rewrite.log estará escrito en el subdirectorío que se encuentra bajo el directorío ServerRoot. El nivel de registro está fijado en 5 para incluir una cantidad razonable de información. Suponiendo que su directiva DocumentRoot se encuentra asignada en /usr/local/apache/htdocs y ServerRoot en /usr/local/apache, necesitará utilizar las siguientes directivas específicas para reglas de escritura en su httpd.conf.

```
RewriteLog logs/rewrite.log  
RewriteLogLevel 5  
  
RewriteCond           /usr/local/apache/htdocs/  
dir1 %{REQUEST_FILENAME} -f  
RewriteRule ^(.+)    /usr/local/apache/htdocs/dir1$1 [L]  
  
RewriteCond           /usr/local/apache/htdocs/  
dir2 %{REQUEST_FILENAME} -f  
RewriteRule ^(.+)    /usr/local/apache/htdocs/dir2$1 [L]  
  
RewriteRule ^(.+)    - [PT]
```

Una vez que ha reiniciado Apache, haga lo siguiente:

1. Ejecute, como raíz, tail -f /usr/local/apache/logs/rewrite.log. Este comando le permite ver entradas de registro a medida que se añaden al archivo rewrite.log.
2. Ejecute mkdir -p /usr/local/apache/htdocs/dir1 y chmod -R httpd:httpd /usr/local/apache/htdocs/dir1 para crear el directorio dir1 bajo su documento y para cambiar su dueño a usuario y grupo httpd. Estoy suponiendo que ha ejecutado Apache como el usuario httpd. Haga lo mismo para dir2.
3. Ejecute el comando lynx -dump -head http://localhost/kabir.html. Esto lanzará el navegador Web Lynx y le dirá que muestre

únicamente las cabeceras de respuesta devueltas por el servidor Web. Ahora, suponiendo que no tiene un archivo llamado `Kabir.html` en el directorio `/usr/local/apache/dir1`, ni en el `/usr/local/apache/dir2`, ni en el `/usr/local/apache`, debería ver una respuesta parecida a la siguiente:

```
HTTP/1.1 404 Not Found
Date: Fri, 16 Mar 2001 06:06:51 GMT
Server: Apache/2.0.14 (Unix)
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

4. A continuación podemos observar las siguientes entradas `rewrite.log`. He borrado las direcciones IP, el temporizador y otros campos, por cuestiones de brevedad.

```
(2) init rewrite engine with requested uri /kabir.html
(3) applying pattern '^(.+)' to uri '/kabir.html'
(4) RewriteCond: input='/usr/local/apache/htdocs/dir1/
kabir.html' pattern='-f' => not-matched
(3) applying pattern '^(.+)' to uri '/kabir.html'
(4) RewriteCond: input='/usr/local/apache/htdocs/dir2/
kabir.html' pattern='-f' => not-matched
(3) applying pattern '^(.+)' to uri '/kabir.html'
(2) forcing '/kabir.html' to get passed through to next API
URI-to-filename handler
```

**NOTA:** Observe cómo `mod_rewrite` intentó localizar el archivo `kabir.html` en `dir1` y en `dir2` y cómo después se riñó, lo que dio lugar a una búsqueda en Apache del mismo archivo en la raíz de documentos porque la solicitud fue `http://localhost/kabir.html`.

5. A continuación ha de crear el archivo de prueba, `kabir.html`, en el subdirectorío `dir1` o en el subdirectorío `dir2`, y cambiar el dueño del archivo de modo que Apache (usuario `httpd`) pueda leerlo. Entonces ejecute el mismo comando `lynx -dump -head http://localhost/kabir.html` de nuevo, y mire el contenido de `rewrite.log`. Verá que una de las reglas ha tenido éxito basándose en el lugar (`dir1` o `dir2`) en el que colocó el archivo.
6. Si tiene el archivo de prueba en los tres directorios, `/usr/local/apache/htdocs/dir1`, `/usr/local/apache/htdocs/dir2` y `/usr/local/apache/htdocs`, la regla de reescritura elige el archivo del subdirectorío `dir1` porque gana la primera condición de coincidencia como consecuencia del indicador `[L]` (last).

## Asignar una variable de entorno basándose en una URL

Es posible que quiera mantener información de estado entre solicitudes y utilizar la URL para codificarla. Pero puede evitar utilizar un script CGI para todas las páginas con el fin de sacar esta información. Puede utilizar una regla de reescritura para obtener la información de estado y almacenarla mediante una variable de entorno que más tarde se puede desreferenciar desde XSSI o desde CGI. De este modo, se traduce una URL /foo/S=java/bar/ a /foo/bar/ y a la variable de entorno llamada STATUS se le asigna el valor java. La figura 9.5 ilustra lo que está ocurriendo.

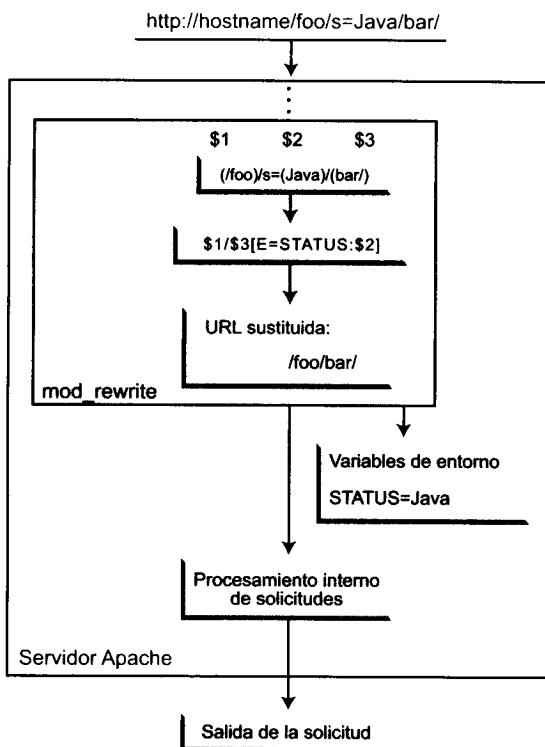


Figura 9.5. Asignar una variable de estado desde una URL

Cuando se detecta una solicitud para `http://hostname/foo/s=java/bar/` el servidor Apache asigna \$1 a foo, \$2 a java, y \$3 a bar, tal y como se muestra en la figura 9.5. El valor para \$2 se utiliza para asignar la variable de entorno STATUS y el servidor Web lleva a cabo un redireccionamiento interno a la localización /foo/bar. Los pasos de reescritura de URL mostrados en la figura 9.5 se pueden implementar utilizando esta regla de reescritura:

```
RewriteRule ^(.*)/S=([^/]+)/(.*) $1/$3 [E=STATUS:$2]
```

El valor de \$2 se almacena en la variable de entorno llamada STATUS utilizando el indicador E. Cuando esta regla está en su sitio y se realiza una solicitud del tipo lynx -dump -head http://localhost/dir1/S=value/kabir.html, el archivo de reescritura (si está disponible) mostrará:

```
(2) init rewrite engine with requested uri /dir1/S=value/
kabir.html
(3) applying pattern '^(.*)/S=([^\/]+)/(.*)' to uri '/dir1/
S=value/kabir.html'
(2) rewrite /dir1/S=value/kabir.html -> /dir1/kabir.html
(5) setting env variable 'STATUS' to 'value'
(2) local path result: /dir1/kabir.html
(2) prefixed with document_root to /usr/local/apache/htdocs/
dir1/kabir.html
(1) go-ahead with /usr/local/apache/htdocs/dir1/kabir.html [OK]
```

He acortado la salida de registro de reescritura por motivos de brevedad. Observe que mod\_rewrite ha fijado la variable STATUS en 'value', por lo que ahora si un script CGI basado en Perl quiere acceder al valor de la variable de entorno STATUS, puede utilizar \$ENV{STATUS}. De igual modo, una directiva Server-Side Include (SSI) también puede acceder a esta variable de entorno.

## Crear sitios www.username.domain.com

Vamos a imaginar que tiene unos cuantos amigos que quieren sitios Web en su servidor. En lugar de darles el sitio del tipo http://www.domain.com/~username (http://www.dominio.com/nombredeusuario), puede crear sitios del tipo http://www.username.domain.com (http://www.nombredeusuario.com) para cada uno de ellos. Naturalmente, tiene que añadir cada nombre de host basado en el nombre de usuario (por ejemplo, www.kabir.domain.com) en su DNS, utilizando un registro CNAME que está dirigido a su servidor Web. Por ejemplo, en su DNS podría tener:

www.domain.com.	IN	A	192.168.1.100.
-----------------	----	---	----------------

Para crear un sitio Web para dos amigos llamados Joe y Jennifer, necesita los siguientes registros DNS para domain.com:

www.domain.com.	IN	A	192.168.1.100.
www.joe.domain.com.	IN	CNAME	www.domain.com.
www.jennifer.domain.com.	IN	CNAME	www.domain.com.

Una vez que el DNS está preparado y probado, tendrá que configurar Apache para servir estos sitios utilizando directorios /home/username/www, en los que cada username es el nombre de la cuenta de usuario de su amigo.

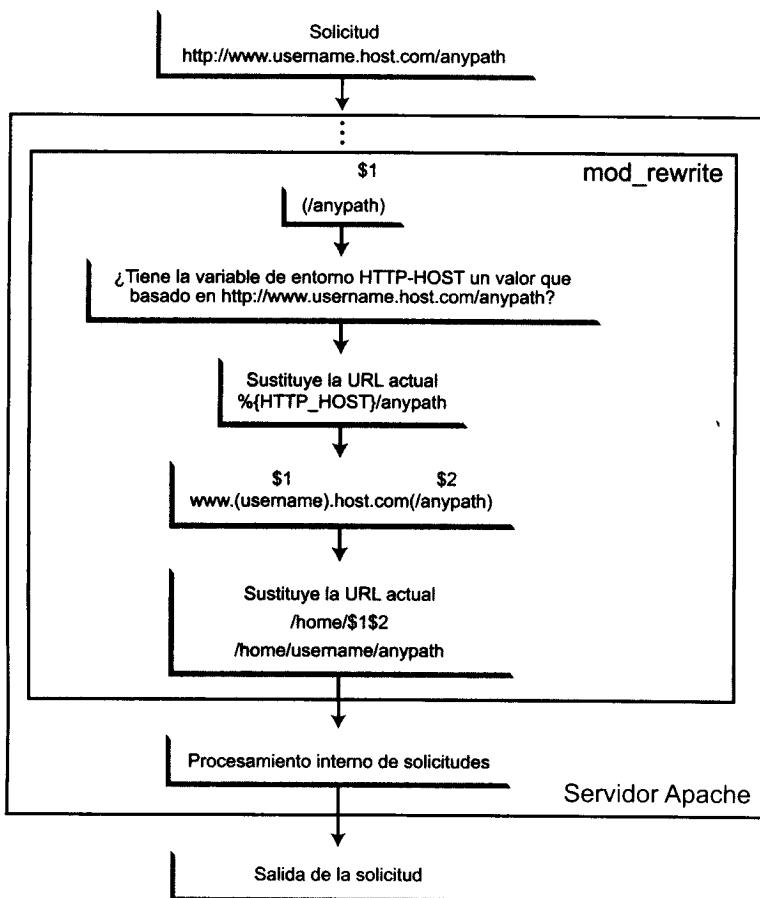
La siguiente asignación de la regla de reescritura se puede utilizar para reescribir internamente http://www.username.domain.com/anypath a /home/username/www/anypath:

```

RewriteCond %{HTTP_HOST} ^www\.[^.]+\.\domain\.com$
RewriteRule ^(.+)$ %{HTTP_HOST}$1 [C]
RewriteRule ^www\.(^.+)\.\domain\.com(.*) /home/$1/www/$2

```

La figura 9.6 ilustra este trabajo.



**Figura 9.6.** Host virtual para cada username

Este es un ejemplo de un conjunto de reglas encadenadas. La primera regla tiene una condición que comprueba si la variable de entorno `HTTP_HOST` tiene un patrón del tipo `www.username.domain.com`. En caso de tenerlo, se aplica la regla. En otras palabras, `www.username.domain.com/anypath` se sustituye por una solicitud del tipo `http://www.username.domain.com/anypath`. Esto puede resultar algo confuso porque la sustitución no es demasiado obvia. Esta sustitución es necesaria por lo que se puede extraer el nombre de usuario (`username`) utilizando la segunda regla. Ésta extrae la parte del nombre de usuario de la solicitud sustituida y crea una nueva URL `/home/username/www/anypath` para una subsolicitud interna.

# Redireccionar una URL fallida a otro servidor Web

Si tiene una red de varios servidores Web y mueve contenido de uno a otro con cierta frecuencia, se podría enfrentar a la necesidad de redirigir solicitudes URL fallidas de un servidor Web A a un servidor Web B. Existen varias formas de hacerlo: puede utilizar la directiva `ErrorDocument`, escribir un script CGI o utilizar `mod_rewrite` para rescribir las URL fallidas en el otro servidor. Utilizar la solución basada en `mod_rewrite` es menos adecuado que utilizar una directiva `ErrorDocument` o un script CGI. La solución `mod_rewrite` tiene el mejor rendimiento, pero es menos flexible y segura en cuanto a errores:

```
RewriteCond /your/docroot/%{REQUEST_FILENAME} !-f  
RewriteRule ^(.+)$ http://Web serverB.dom/$1
```

El problema es que esta solución únicamente funcionará para páginas en el interior de la directiva `DocumentRoot`. Aunque pueda añadir más condiciones (para manejar directorios home, por poner un ejemplo), hay una variante mejor:

```
RewriteCond %{REQUEST_URI} !-U  
RewriteRule ^(.+)$ http://Web serverB.dom/$1
```

Esta variante utiliza la característica URL de proyecto de `mod_rewrite`, y funcionará en todos los tipos de URL. Esto va a tener un impacto de rendimiento en el servidor Web porque para cada solicitud realizada, hay una subsolicitud interna. Utilice esta opción si su servidor Web se ejecuta en una CPU potente; si se trata de una máquina lenta, utilice la primera aproximación, o, aún mejor, una directiva `ErrorDocument` o un script CGI.

## Crear un acceso multiplexor

Este ejemplo le muestra cómo crear un conjunto de reglas para redirigir solicitudes basadas en un tipo de dominio, como `.com`, `.net`, `.edu`, `.org`, `.uk`, `.de`, y similares. La idea es redirigir al visitante al sitio Web geográficamente más cercano. Las grandes corporaciones emplean esta técnica para redireccionar a los clientes internacionales al sitio Web o al servidor FTP adecuado.

El primer paso para crear una solución de este tipo es crear un archivo `map`. El siguiente ejemplo muestra un archivo `map` basado en texto llamado `site-redirect.map`:

com	http://www.mydomain.com/download/
net	http://www.mydomain.com/download/
edu	http://www.mydomain.com/download/
org	http://www.mydomain.com/download/
uk	http://www.mydomain.uk/download/

de                    <http://www.mydomain.de/download/>  
ch                    <http://www.mydomain.ch/download/>

Cuando se recibe una solicitud para <http://www.mydomain.com/download/anypath> desde un host llamado `dialup001.demon.uk`, la solicitud necesita ser redirigida al sitio Web `www.mydomain.uk/download/`; del mismo modo, cualquier solicitud desde un host que pertenezca a los dominios de nivel máximo `.com`, `.net`, `.edu` y `.org` son enrutados al sitio `www.mycompany.com/download/`.

Aquí tenemos las reglas que se necesitan para el sistema anterior:

```
RewriteMap sitemap txt:/path/to/site-redirect.map  
RewriteRule ^/download/(.*) %{REMOTE_HOST}:::$1 [C]  
RewriteRule ^.+\.([a-zA-Z]+)::(.*)$  
  %{sitemap:$1|www.mydomain.com/download/}$2 [R,L]
```

La figura 9.7 ilustra la utilización de esta regla.

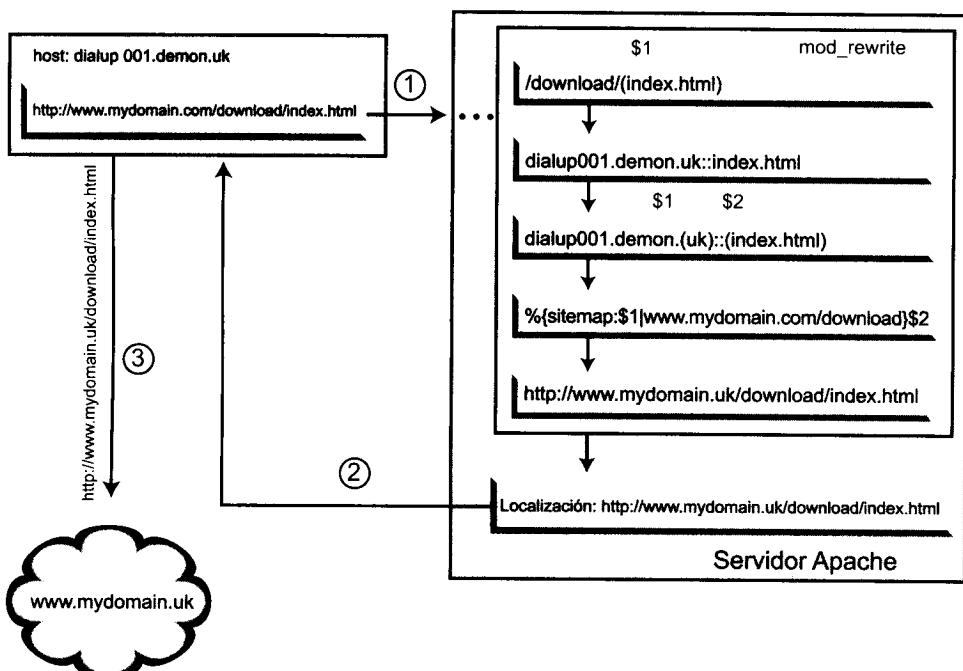


Figura 9.7. Un acceso multiplexor basado en URL

Tal y como muestra la figura 9.7, cuando un host como `dialup001.demon.uk` solicita la página `www.mydomain.com/download/index.html` (1), la primera regla reescribe la solicitud utilizando el nombre del host solicitado:

`dialup001.demon.uk::index.html`

Entonces se aplica la siguiente regla de la cadena. Esta regla se aplica cuando el patrón de búsqueda coincide y se crea la URL sustituida colocando el archivo map en el dominio de máximo nivel. Si no se encuentran coincidencias, se utiliza `www.mydomain.com` que es la URL por defecto. Esto lo lleva a cabo el operador | (or) en la cadena URL de sustitución. Quizá es más fácil de entender la segunda regla utilizando el algoritmo tal y como se muestra en el listado 9.1.

#### Listado 9.1. Algoritmo para la segunda regla de reescritura

```
if(current URL matches a fully-qualified-hostname::anything)
then

# substituye la URL actual utilizando la información del tipo de
# dominio, almacenada en $1, realizando una búsqueda en el
archivo map.

If (map file has a key that matches the domain type) then
#utiliza el valor de la clave del siguiente modo:

    Substituted URL = value-of-the-key$2
    #donde $2 es cualquier cosa que se encuentre después de
    #fully-qualified-hostname:: pattern

Else
    # Utiliza el valor por defecto www.mydomain.com/
download/$2

endif

Substituted URL = www.mydomain.com/download/$2
endif

Endif
```

El indicador R lo convierte en una redirección URL externa y el indicador L lo convierte en la última regla para la URL sustituida. La nueva localización creada por la regla #2 se envía al navegador Web (2) y el navegador obtiene la página en (3) tal y como se muestra en la figura 9.7.

## Crear URL dependientes del tiempo

Es posible que alguna vez se haya preguntado si se puede tener una URL que pudiese dirigirse a distintos archivos dependiendo del momento. `mod_rewrite` crea de un modo sencillo este tipo de URL. Hay muchas variables llamadas `TIME_xxx` para reescribir condiciones. Utilizando los patrones de comparación especiales `<STRING>`, `=STRING` y `>STRING` puede realizar redireccionamientos dependientes del tiempo, por ejemplo:

```
RewriteCond %{TIME_HOUR}%{TIME_MIN} >0700
RewriteCond %{TIME_HOUR}%{TIME_MIN} <1900
```

```
RewriteRule      ^foo\.html$          foo.day.html
RewriteRule      ^foo\.html$          foo.night.html
```

Esto proporciona el contenido de `foo.day.html` bajo la URL `foo.html` desde 07:00 a 19:00, y el resto del tiempo, proporciona el contenido de `foo.night.html`.

## Manejar contenido

Los ejemplos de esta sección versan sobre reglas de reescritura de contenido específico. Le mostraré cómo crear las URL de compatibilidad retroactiva, a reescribir contenido basado en el navegador, a crear HTML transparente al usuario final para las redirecciones CGI, y este tipo de cosas.

### Añadir compatibilidad retroactiva en las URL

Imagine que ha renombrado recientemente la página `bar.html` a `foo.html` y que ahora quiere proporcionar la antigua URL para tener compatibilidad retroactiva. Además, no quiere que los usuarios de la antigua URL se den cuenta de que la página ha sido renombrada. ¿Cómo puede hacer esto? A continuación le mostramos la manera:

```
RewriteRule      ^foo\.html$    bar.html
```

Si quiere dejar que el navegador se dé cuenta del cambio, puede realizar una reescritura externa de modo que el navegador mostrará la nueva URL. Todo lo que necesita hacer es añadir el indicador R del siguiente modo:

```
RewriteRule      ^foo\.html$    bar.html [R]
```

### Crear las URL con contenido específico para el navegador

Puede utilizar reglas de reescritura para servir distinto contenido (utilizando subsolicitudes internas) a distintos navegadores. No puede utilizar negociación de contenido para esto, porque los navegadores no proporcionan sus tipos de este modo. En lugar de esto, tiene que actuar en la cabecera HTTP User-Agent. Por ejemplo, si la cabecera User-Agent de un navegador coincide con Mozilla/5 entonces puede enviar una página de características afines a Netscape Navigator 5 (o superior), o puede enviar una página distinta si el navegador es una versión más antigua de Navigator u otro tipo de navegador.

Si la cabecera HTTP User-Agent empieza por Mozilla/5, la página `foo.html` es reescrita a `foo.NS.html` y termina la reescritura. Si el navegador es Lynx o

la versión de 1 a 4 de Mozilla, la URL se convierte en `foo.dull.html`. El resto de los navegadores reciben la página `foo.cool.html`. Esto se lleva a cabo con las reglas siguientes:

```
RewriteCond %{HTTP_USER_AGENT} ^Mozilla/5.*  
RewriteRule ^foo\.html$ foo.cool.html [L]  
  
RewriteCond %{HTTP_USER_AGENT} ^Lynx/.*/[OR]  
RewriteCond %{HTTP_USER_AGENT} ^Mozilla/[1234].*  
RewriteRule ^foo\.html$ foo.dull.html [L]
```

Cuando se recibe una solicitud de una URL del tipo `http://hostname/foo.html`, la primera condición comprueba si la variable de entorno `HTTP_USER_AGENT` tiene un valor que contiene la cadena `Mozilla/5.` o no lo tiene. Si contiene la cadena, entonces se aplica la primera regla. Esta regla sustituye `foo.cool.html` por la URL original y se completa la reescritura; sin embargo, cuando no se aplica la primera regla, se invoca la segunda. Hay dos condiciones OR. En otras palabras, una de estas condiciones deben coincidir antes de que esta regla se aplique.

La primera condición prueba la misma variable de entorno para la subcadena `Lynx/`, y la segunda condición prueba la misma variable de entorno para la subcadena `Mozilla/1` a través de `Mozilla/4`. Si se cumple alguna de estas condiciones, se aplica la regla. La regla sustituye `foo.dull.html`, la URL original. La URL sustituida se convierte en una subsolicitud y Apache la procesa de la forma habitual.

## Crear HTML para un puente CGI

Si quiere transformar la página estática `foo.html` en una variante dinámica llamada `foo.cgi`, sin informar ni al navegador ni al usuario, a continuación le mostramos cómo hacerlo:

```
RewriteRule ^foo\.html$ foo.cgi [T=application/x-httpd-cgi]
```

La regla describe una solicitud para `foo.html` a una solicitud para `foo.cgi`. Además fuerza el tipo correcto de MIME, de modo que se ejecuta como un script CGI. Una solicitud del tipo `http://hostname/foo.html` se traduce internamente en una solicitud para el script CGI. El navegador no sabe que esta solicitud se ha redirigido.

## Restricción de acceso

Estos ejemplos muestran temas sobre el control de acceso. En esta sección le muestro cómo controlar el acceso a ciertas áreas de su sitio Web utilizando el módulo para rescribir URL.

# Robots de bloqueo

Es sencillo bloquear un programa rastreador de Web (también llamados robots) molesto, para que no pueda obtener páginas de un sitio Web determinado. Puede probar con un archivo `/robots.txt` que contenga entradas de Robot Exclusion Protocol, pero, normalmente, esto no es suficiente para deshacerse de estos tipos de robots. Una solución podría ser:

```
RewriteCond %{HTTP_USER_AGENT}      ^NameOfBadRobot.*  
RewriteCond %{REMOTE_ADDR}          ^123\.45\.67\.[8?9]$  
RewriteRule ^/not/to/be/indexed/by/robots/.+ - [F]
```

Esta regla tiene dos condiciones:

```
If  (HTTP_USER_AGENT of the robot matches a pattern  
"NameOfBadRobot" ) and  
    (REMOTE_ADDR of the requesting host is 123.45.67.8 to  
123.45.67.9) then  
    No substitution but send a HTTP "Forbidden" header (status  
code 403)  
endif
```

Como puede ver, coincide la cabecera User-Agent del robot, junto con la dirección IP del host que se utiliza. Las condiciones anteriores permiten que se comprueben varias direcciones IP (123.45.67.8 y 123.45.67.9).

## Crear deflector URL basado en una referencia HTTP

Puede programar un deflector URL flexible que actúe en la cabecera Referer HTTP y que la configure con tantas páginas como quiera. A continuación tenemos el modo de hacerlo:

```
RewriteMap deflector txt:/path/to/deflector.map  
RewriteRule ^/(.*)  
${deflector:%{HTTP_REFERER}|/$1}  
RewriteRule ^/DEFLECTED    %{HTTP_REFERER} [R,L]  
RewriteRule .* - [PT]
```

Esto se utiliza junto con el mapa correspondiente de reescritura:

```
http://www.badguys.com/bad/index.html    DEFLECTED  
http://www.badguys.com/bad/index2.html   DEFLECTED  
http://www.badguys.com/bad/index3.html   http://somewhere.com/
```

Esto redirecciona automáticamente las solicitudes a la página de referencia si la URL coincide con el valor DEFLECTED en el archivo map. En todos los casos, las solicitudes se redireccionan a las URL especificadas.





# **10 Establecer un servidor Proxy**

---

## **En este capítulo**

1. Analizamos los tipos de servidores proxy.
2. Configuramos Apache como un servidor proxy.
3. Preparamos navegadores Web para proxies.

Un servidor proxy es un sistema que se sitúa entre el host del cliente y el servidor al que quiere acceder. Cuando un host solicita un cierto recurso remoto utilizando una URL, el servidor proxy recibe esta solicitud y utiliza el recurso para satisfacer la solicitud del cliente. En términos generales, un servidor proxy actúa como un servidor para los host del cliente y como un cliente para los servidores remotos.

En un escenario típico de proxy, este proceso permite al servidor proxy almacenar el contenido solicitado en un caché. Cualquier solicitud nueva que requiera información que se encuentra en el caché no necesita ser servida trayendo la información desde el servidor remoto. En lugar de eso, la nueva solicitud se sirve desde los datos del caché. Esto permite a los servidores proxy suavizar los cuellos de botella. Sin embargo, esto no es todo lo que un servidor proxy puede hacer. Este capítulo le enseña a convertir Apache en un servidor proxy que puede llevar

a cabo multitud de servicios. Aprenderá a convertir Apache en un servidor proxy de caching (forward). Distribuir este tipo de servidores en un cuello de botella puede reducir retrasos en los tiempos de respuesta, conservar el ancho de banda y, además, le ayuda a reducir el gasto general de sus comunicaciones. Como el proxy se utiliza, normalmente, para redes con grandes comunidades de usuarios, también vamos a tratar los distintos aspectos de la configuración automática del proxy.

## ¿Quién debería utilizar un servidor proxy?

El propósito de este servidor proxy es traer los recursos solicitados desde el servidor remoto, devolverlos al usuario que los solicita y cachearlos en los drivers locales. El servicio proxy es perfecto para escenarios en los que están accediendo a la red varios usuarios. Muchas organizaciones tienen varios ordenadores host que acceden a Internet mediante una sola conexión Internet, como un router RDSI o cualquier otra conexión dedicada o en demanda. Un proxy puede ser muy útil en tales redes. Utilizando un proxy tanto para Internet como para su intranet puede obtener los siguientes beneficios:

- **Proxying:** si la red interna utiliza direcciones IP no ruteables por razones de seguridad o económicas, puede utilizar un servidor proxy para proporcionar recursos de Internet a los host que normalmente no pueden acceder a Internet. Este capítulo le enseña cómo hacerlo.
- **Caching:** utilizando un proxy caching como Apache (con mod\_perl), puede proporcionar a los usuarios locales de los recursos de Internet un acceso más rápido. Esto no sólo aumentará el rendimiento de la red según la percepción del usuario sino que, además, reducirá los costes de utilización de ancho de banda.
- **Control de registro y acceso:** utilizando un servidor proxy, puede controlar el uso que hacen los empleados o los estudiantes de Internet (o incluso de intranet). Puede bloquear el acceso a determinados sitios Web para proteger a su compañía, y puede evitar que abusen del tiempo de su compañía. Analizando los accesos a su servidor proxy y los registros de error, puede identificar los patrones de uso y realizar una política de utilización de la red más adecuada en el futuro.

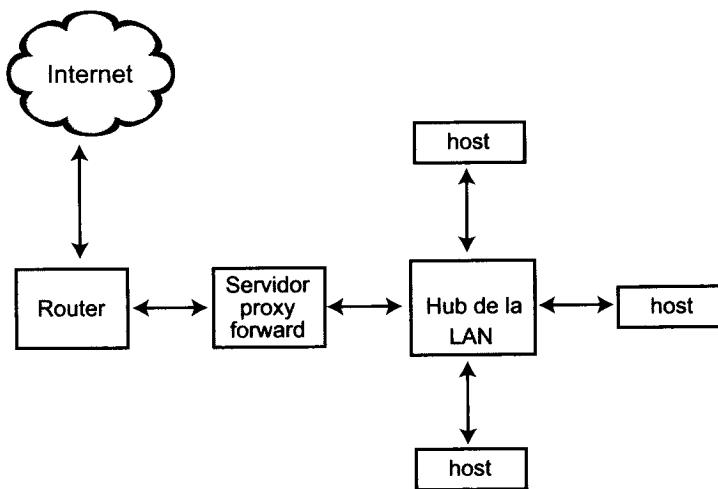
## Análisis de los tipos de servidores proxy

Antes de hablar de la utilización de Apache como servidor proxy, vamos a discutir los tipos de servidores proxy y su funcionamiento. Hay dos tipos de servidores proxy:

- **Servidores proxy forward:** cuando se utiliza este tipo de servidores proxy, los usuarios pasan sus solicitudes al servidor proxy y éste obtiene la respuesta del host objetivo de la solicitud. Los proxies forward están normalmente definidos explícitamente en los programas usuarios (como en el navegador Web).
- **Servidores proxy reverse:** cuando se utiliza este tipo de servidor, los usuarios no son conscientes de su existencia porque piensan que están accediendo al recurso directamente. Todas las solicitudes que realiza el usuario se envían al proxy reverse, que sirve la respuesta desde su caché o recogiendo información de otro host.

## Proxy forward

Un proxy forward normalmente se sitúa entre el host del usuario y los recursos remotos a los que quiere acceder. Un recurso puede ser un recurso de Internet, como muestra la figura 10.1, o puede ser un recurso de intranet. La siguiente solicitud para el mismo recurso será servida desde los datos del caché si los datos no han expirado.



**Figura 10.1.** Un servidor proxy forward

Los hosts usuarios saben que están utilizando un servidor proxy porque cada host debe ser configurado para utilizar un servidor proxy. Así por ejemplo, es necesario que le diga al navegador Web que utilice un servidor proxy antes de que el navegador pueda utilizarlo. Todas las solicitudes remotas están encauzadas mediante el servidor forward mostrado en la figura 10.1, proporcionando, de este modo, una solución manejable y de coste efectivo para reducir la utilización de ancho de banda e implementar la política de acceso a usuarios. Este tipo de servidores proxy también se conocen con el nombre de *servidor proxy caching*. El

proxy reverse también mete datos en el caché pero funciona de forma contraria a como lo hace el servidor proxy forward.

## Proxy reverse

Un proxy reverse se sitúa frente a un recurso de Internet, tal y como muestra la figura 10.2, o frente a un recurso de intranet. En estos sistemas, el proxy reverse recupera los recursos solicitados del servidor original y los devuelve al host usuario.

El host usuario conectado con el servidor proxy no es consciente de estar conectado al servidor proxy, sino que piensa que está conectado directamente al servidor que contiene los recursos, a diferencia de lo que ocurre cuando utilizamos un servidor forward. Desde el punto de vista del usuario, está accediendo al recurso solicitado directamente.

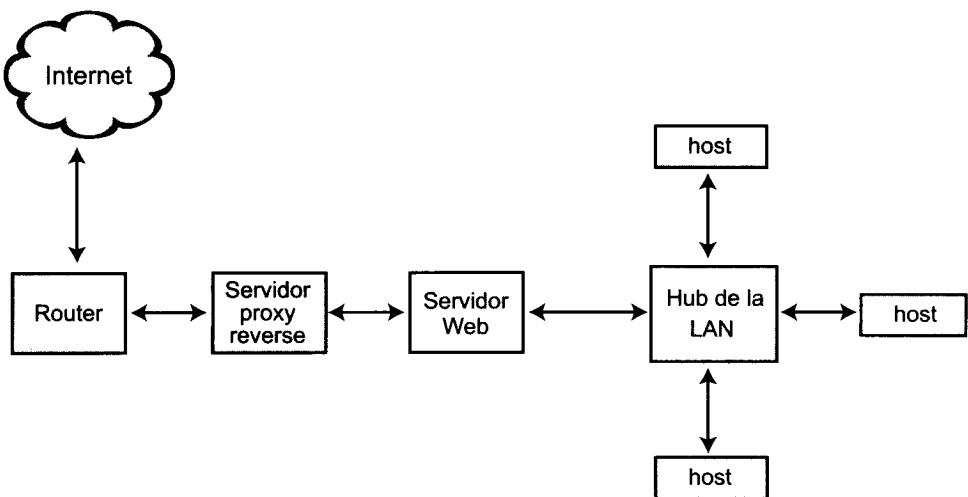


Figura 10.2. Un servidor proxy reverse

La figura 10.2 muestra que los usuarios de Internet tienen que atravesar el proxy reverse para conectarse al servidor Web. Como es un proxy reverse, los usuarios no son conscientes de su existencia y piensan que están conectados con el servidor Web. Los usuarios de una LAN pueden conectar directamente con el servidor Web desde la red interna, tal y como se muestra en la figura.

Por ejemplo, si se utiliza un servidor proxy reverse para un sitio Web llamado [www.csus.edu](http://www.csus.edu), entonces todos los estudiantes de CSUS (California State University, Sacramento) tienen que dirigir su navegador a [www.csus.edu](http://www.csus.edu) y no hacer referencia a ninguna configuración proxy en su navegador. El navegador manda una solicitud al servidor [www.csus.edu](http://www.csus.edu). Lo cierto es que al navegador no le sirve de mucho saber que el servidor [www.csus.edu](http://www.csus.edu) es en realidad un servidor proxy reverse que traduce internamente las solicitudes a un servidor Web.

llamado internal-www.csus.edu, para obtener el contenido de la solicitud. ¿Qué ventaja tiene este tipo de sistemas? Como el dato está cacheado para cada solicitud, el servidor proxy puede proporcionar algún soporte de balance de carga para los verdaderos servidores que se encuentran detrás de la escena.

**NOTA:** Apache no soporta por ahora el servicio proxy reverse; sin embargo, se implementará en la siguiente versión del módulo mod\_proxy.

## Directivas mod\_proxy

El soporte proxy en Apache proviene del módulo mod\_proxy. Este módulo no está compilado por defecto. Tiene que reconfigurar Apache utilizando ./config.status --enable-module=proxy del directorio de la distribución de la fuente de Apache y ejecutar make && make install para recomilar y reinstalar el servidor Apache actualizado. Actualmente, sólo está implementado un servidor proxy caching en Apache. Es capaz de soportar los protocolos HTTP 1.1, HTTPS (vía CONNECT para SSL) y FTP. El módulo también se puede configurar para conectarse a otros módulos proxy para estos y otros protocolos. Proporciona las directivas que se discuten en las siguientes secciones.

### ProxyRequests

ProxyRequests le permite activar o desactivar el servicio caching del proxy. Sin embargo, no afecta a la funcionalidad de la directiva ProxyPass.

**Sintaxis:** ProxyRequests On | Off

**Predefinido:** ProxyRequests Off

**Contexto:** configuración del servidor, host virtual

### ProxyRemote

ProxyRemote le permite a su servidor funcionar de interfaz con otro servidor proxy.

**Sintaxis:** ProxyRemote match remote\_proxy\_server\_URL

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

El valor de correspondencia puede ser alguno de los siguientes:

- El nombre de un esquema URL que soporta el servidor remoto.
- Una URL parcial para la que se pueda utilizar el servidor remoto.
- Para indicar el servidor que debería ser contactado para todas las solicitudes.

`remote_proxy_server_URL` puede tener un valor del tipo `http://hostname-:port`. Observe que, por ahora, únicamente se soporta el protocolo HTTP. En otras palabras, puede especificar únicamente un servidor proxy que trate con el protocolo HTTP; sin embargo, puede dirigir las solicitudes FTP desde su servidor proxy a uno que soporte los protocolos HTTP y FTP del siguiente modo:

```
ProxyRemote ftp http://ftp.proxy.evoknow.com:8000
```

envía todas las solicitudes FTP que proceden del servidor proxy local a `ftp://ftp.proxy.evoknow.com`. Las solicitudes se envían vía HTTP, por lo tanto, la verdadera transacción FTP tiene lugar en el servidor proxy remoto.

Si quiere dirigir todas las solicitudes proxy de un determinado sitio Web a este servidor proxy directamente, puede hacerlo con esta directiva. Por ejemplo:

```
ProxyRemote http://www.bigisp.com/ http://web-proxy.bigisp.com:8000
```

Esto envía todas las solicitudes que coinciden con `www.bigisp.com` a `web-proxy.bigisp.com`. Si quiere dirigir todas sus solicitudes proxy a otro proxy, puede utilizar el asterisco como correspondencia. Por ejemplo:

```
ProxyRemote * http://proxy.domain.com
```

envía todas las solicitudes proxy locales al servidor proxy `proxy.domain.com`.

## ProxyPass

`ProxyPass` le permite mapear un árbol de documentos del servidor Web en el espacio de documentos de su servidor proxy.

**Sintaxis:** `ProxyPass relative_URL destination_URL`

**Contexto:** configuración del servidor, host virtual

Por ejemplo:

```
ProxyPass /internet/microsoft www.microsoft.com/
```

Si `ProxyPass` se encuentra en el archivo `httpd.conf` de un servidor proxy llamado `proxy.evoknow`, permitirá a los usuarios del servidor proxy que accedan al sitio Web de Microsoft utilizando la URL:

<http://proxy.evoknow.com/internet/microsoft>

actúa como un mirror del sitio Web remoto. Cualquier solicitud que utiliza el <relative-URL> se convertirá internamente en una solicitud proxy para él <destination-URL>.

Si el sitio remoto incluye referencias absolutas, las imágenes no aparecerán y los enlaces no funcionarán. Tampoco podrá utilizar esta directiva con los servidores SSL.

## ProxyBlock

La directiva ProxyBlock le permite bloquear el acceso al host o al dominio.

**Sintaxis:** ProxyBlock partial\_or\_full\_hostname [ . . . ]

**Contexto:** configuración del servidor, host virtual

Por ejemplo:

```
ProxyBlock gates
```

bloquea el acceso a cualquier host que tenga la palabra gates en su nombre. De este modo, el acceso a `http://gates.ms.com` o a `http://gates.friendsofbill.com` está bloqueado. También puede especificar varios hosts.

Por ejemplo:

```
ProxyBlock apple orange.com bannana.com
```

bloquea todos los accesos a cualquier host que coincida con cualquiera de las palabras o nombres de dominio anterior. El módulo mod\_proxy intenta determinar la dirección IP para estos hosts durante el arranque del servidor, y los cachea para correspondencias posteriores.

Para bloquear el acceso a todos los hosts, utilice:

```
ProxyBlock *
```

Esto efectivamente inutiliza su servidor proxy.

## NoProxy

NoProxy le proporciona cierto control sobre la directiva ProxyRemote en un entorno de intranet.

**Sintaxis:** NoProxy Domain\_name | Subnet | IP\_Address | Hostname

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

Puede especificar que un nombre de dominio, o una subred, o una dirección IP, o un nombre de host no sea servido por el servidor proxy especificado en la directiva ProxyRemote. Por ejemplo:

```
ProxyRemote * http://firewall.yourcompany.com:8080  
NoProxy .yourcompany.com
```

Aquí todas las solicitudes para <anything>.yourcompany.com (como www.yourcompany.com) están servidas por el servidor proxy local y el resto se dirigen al servidor proxy firewall.yourcompany.com.

## ProxyDomain

ProxyDomain especifica el nombre de dominio para el servidor proxy.

**Sintaxis:** ProxyDomain Domain

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

Cuando esta directiva tiene asignado el nombre de dominio local en una intranet, cualquier solicitud que no incluye a un nombre de dominio, tendrá este nombre de dominio adjunto; por ejemplo:

```
ProxyDomain .evoknow.com
```

Cuando un usuario del dominio evoknow.com envía una solicitud para una URL del tipo http://marketing/us.html, la solicitud es regenerada como la siguiente URL:

```
http://marketing.evoknow.com/us.html
```

Observe que el nombre de dominio que se especifica en la directiva ProxyDomain debe tener un punto inicial.

## CacheRoot

CacheRoot le permite cachear el disco. Puede especificar un nombre de directorio en el que el servidor proxy puede escribir archivos cacheados.

**Sintaxis:** CacheRoot directory

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

El servidor Apache que está ejecutando el módulo proxy, debe escribir permisos para el directorio, por ejemplo:

```
CacheRoot /www/proxy/cache
```

le dice a Apache que escriba datos cacheados del proxy en el directorio /www/proxy/cache. Observe que necesitará especificar el tamaño del caché utilizando el directorio CacheSize antes de que el servidor proxy pueda empezar a utilizar este directorio para caching. Necesitará utilizar otras directivas caché (que se discutirán más tarde) para crear una solución proxy con un disco cacheable.

## CacheSize

CacheSize determina la cantidad de espacio en el disco (en kilobytes) que se debe utilizar para el cacheado del disco. Los archivos cacheados están escritos en el directorio especificado por la directiva CacheRoot.

**Sintaxis:** CacheSize kilobytes

**Predefinido:** CacheSize 5

**Contexto:** configuración del servidor, host virtual

**NOTA:** Aunque el servidor proxy puede escribir más datos de los que están especificados en el límite, el esquema de la colección basura del servidor proxy eliminará archivos hasta que la utilización se encuentre en el valor asignado o por debajo de éste. La asignación por defecto (5K) no es realista; recomiendo cualquier asignación entre 10MB y 1GB dependiendo de la carga de usuarios.

## CacheGcInterval

CacheGcInterval determina el momento (en horas) en el que Apache debe comprobar los directorios caché para eliminar los archivos que han expirado. Esto también tienen lugar cuando Apache fuerza el límite del espacio de disco utilizado especificado por la directiva CacheSize.

**Sintaxis:** CacheGcInterval hours

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

## CacheMaxExpire

CacheMaxExpire determina el momento (en horas) en el que exiran todos los documentos cacheados. Esta directiva invalida cualquier fecha de expiración especificada en el documento; por lo tanto, si un documento tiene una fecha de expiración posterior al máximo especificado por esta directiva, el documento se elimina igualmente.

**Sintaxis:** CacheMaxExpire hours

**Predefinido:** CacheMaxExpire 24

**Contexto:** configuración del servidor, host virtual

El valor por defecto permite que los documentos cacheados expiren en 24 horas. Si desea que los documentos expiren más tarde cambie este valor.

## CacheLastModifiedFactor

CacheLastModifiedFactor determina un factor que se utiliza para calcular el momento de expiración cuando el servidor Web original no distribuye una fecha de expiración para el documento.

**Sintaxis:** CacheLastModifiedFactor floating\_point\_number

**Predefinido:** CacheLastModifiedFactor 0.1

**Contexto:** configuración del servidor, host virtual

El cálculo se realiza utilizando la fórmula siguiente:

```
expiry-period = (last modification time for the document) *  
(floating point number)
```

Por lo tanto, si se modificó un documento hace 24 horas, el factor por defecto de 0.1 produce un cálculo de 2.4 horas para la expiración de ese documento. Si el tiempo de expiración calculado es superior al asignado por CacheMaxExpire, el período de expiración que marca CacheMaxExpire tiene prioridad.

## CacheDirLength

Cuando está activado el caching de un disco, Apache crea subdirectorios en el directorio especificado por la directiva CacheRoot. Esta directiva determina el número de caracteres utilizados en la creación de los nombres de los directorios. Realmente no necesita cambiar el valor por defecto de esta directiva. Los usuarios curiosos que quieran saber cómo o por qué se han creado estos directorios, tienen una respuesta simplificada a continuación.

**Sintaxis:** CacheDirLength length

**Predefinido:** CacheDirLength 1

**Contexto:** configuración del servidor, host virtual

Apache utiliza un esquema de digestión cuando crea la ruta y el nombre de archivo para los datos de la URL que se van a cachear. Por ejemplo, cuando tiene activado el caching y accede a una URL (como www.microsoft.com) me-

dianamente su servidor proxy Apache, el servidor digiere esta URL de modo que más tarde puede recuperar datos rápidamente. Esta digestión es del tipo `1YSRxSmB20Q_HkqkTuXeqvw`. Si se utiliza el valor por defecto para las directivas `CacheDirLength` y `CacheDirLevels`, Apache almacena los datos encontrados en `www.microsoft.com` en un archivo llamado:

```
%CacheRoot%/1/Y/S/RRxSmB20Q_HkqkTuXeqvw
```

Aquí `%CacheRoot%` es el directorio especificado por la directiva `CacheRoot`. Los directorios `1/Y/S` se han creado para el valor por defecto de la directiva `CacheDirLevels`. Cuando este documento se solicita de nuevo utilizando la misma URL, Apache solo necesita recalcular la digestión para recuperar la página de la ruta especificada.

## CacheDirLevels

`CacheDirLevels` determina el número de directorios que creará Apache para almacenar los archivos de datos cacheados. Remítase a la sección anterior para obtener la información relacionada.

**Sintaxis:** `CacheDirLevels levels`

**Predefinido:** `CacheDirLevels 3`

**Contexto:** configuración del servidor, host virtual

## CacheDefaultExpire

`CacheDefaultExpire` proporciona el tiempo por defecto (en horas) que se utiliza para expirar un archivo cacheado cuando no se sabe el momento de la última actualización de un archivo. `CacheMaxExpire` no invalida esta asignación.

**Sintaxis:** `CacheDefaultExpire hours`

**Predefinido:** `CacheDefaultExpire 1`

**Contexto:** configuración del servidor, host virtual

## NoCache

**Sintaxis:** `NoCache Domain_name | Subnet | IP_Address | Hostname . . . ]`

**Predefinido:** ninguno

**Contexto:** configuración del servidor, host virtual

La directiva `NoCache` determina una lista de host, nombres de dominio y direcciones IP, separados por espacios, para los que no tiene lugar un cacheado. Esta directiva debería utilizarse para deshabilitar el caching de los servidores Web en una intranet.

Observe que el servidor proxy también relaciona nombres parciales de un host. Si quiere deshabilitar todo el caching, utilice:

```
NoCache *
```

## Configurar un servidor proxy Apache

En esta sección le mostraré cómo configurar Apache (con `mod_proxy`) como un servidor proxy forward. Una vez que tiene el módulo `mod_proxy` compilado en Apache (tal y como se ha discutido), establecer un servidor proxy es muy sencillo.

Para activar el servidor proxy, necesita asignarle el valor `On` al `ProxyRequests` en un archivo `httpd.conf`. Cualquier configuración adicional depende de lo que quiera hacer con el servidor proxy.

Independientemente de lo que decida hacer con él, cualquier directiva, que quiera utilizar para controlar el comportamiento del servidor proxy, debe ir dentro de un contenedor `<Directory . . .>` especial que es de la forma siguiente:

```
<Directory proxy:*>
  .
  .
</Directory>
```

El asterisco es un comodín para la URL solicitada. En otras palabras, cuando el servidor Apache procesa una solicitud para `www.evoknow.com`, lo hace del siguiente modo:

```
<Directory proxy:http://www.evoknow.com/>
  .
  .
</Directory>
```

También puede utilizar el contenedor `<Directory ~ /RE/>`, que utiliza expresiones regulares que le permiten una flexibilidad mayor en la definición de la configuración proxy.

Por ejemplo:

```
<Directory ~ proxy:http://[^:/]+/.*>
  .
  .
</Directory>
```

A continuación vamos a ver unas cuantas configuraciones proxy que se utilizan habitualmente.

## Escenario 1: conectar una IP privada a Internet

En este escenario, sólo hay un ordenador en la red que tiene una dirección IP de Internet ruteable, tal y como se muestra en la figura 10.3. Este ordenador ejecuta el servidor proxy de Apache con `ProxyRequest` fijada en `On`, y no se necesita configuración proxy adicional. El servidor proxy sirve todas las solicitudes.

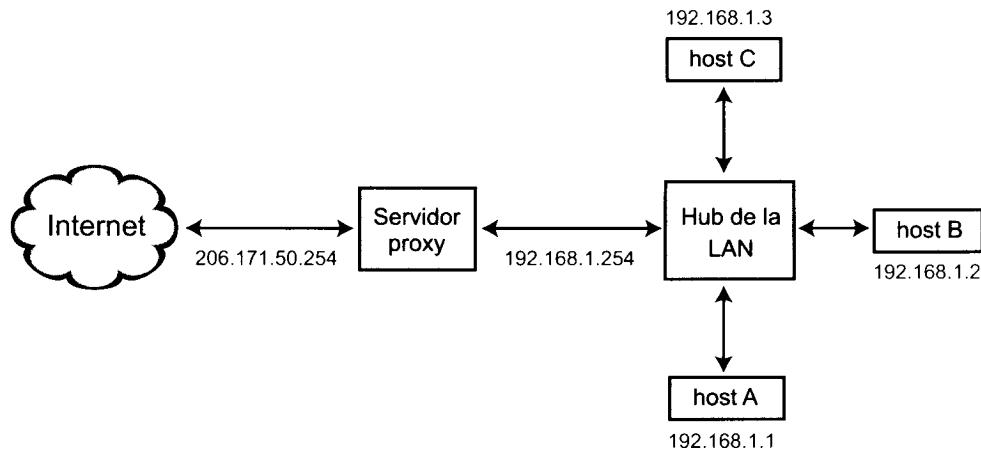


Figura 10.3. Proxy para una red IP privada

En este tipo de configuración, el servicio proxy necesita estar multialojado; en otras palabras, necesita tener acceso a la red privada no ruteable (`192.168.1.0`) y a la red con IP ruteable (`206.171.50.0`). De algún modo, este proxy actúa como un firewall para la red privada aunque ya lo haga el pool de direcciones IP elegido. El proxy permite que el host acceda a los servicios de Internet, como son la Web, FTP, y similares.

## Escenario 2: caching sitios web remotos

Como hay una gran cantidad de contenido Web estático en Internet y en intranet, cachearlo en un servidor proxy local puede ahorrar una gran cantidad de ancho de banda. Un servidor proxy que permite cachear, únicamente sirven los documentos solicitados cuando el caché contiene un documento que ha expirado o cuando el documento solicitado no está en el caché. Para permitir el caching en su servidor proxy, tiene que especificar las directivas de caching dentro de un contenedor de directorios especial. Por ejemplo:

```
<Directory proxy:*>
    CacheRoot /www/cache
    CacheSize 1024
    CacheMaxExpire 24
</Directory>
```

Esta configuración define un servidor proxy de caching que escribe archivos caché en el directorio /www/cache. Está permitido escribir 1024K de datos (1MB) y el caché deberá expirar cada día (24 horas).

Si quiere evitar que alguien abuse de su servidor proxy, puede restringir el acceso mediante autentificación del host o mediante autentificación nombre de usuario/ contraseña.

Para controlar qué hosts tienen acceso al servidor proxy, puede crear una configuración del siguiente tipo:

```
<Directory proxy:>
    AuthType Basic
    AuthName Proxy
    order deny,allow
    deny from all
    allow from myhost.evoknow.com

</Directory>
```

Esta configuración deniega el acceso a todos excepto a myhost.evoknow.com. Si quiere utilizar autentificación nombre de usuario/ contraseña, puede utilizar algo parecido a lo siguiente:

```
<Directory proxy:>
    AuthType Basic
    AuthName Proxy
    AuthUserFile /path/to/proxy/.htpasswd
    AuthName Proxy
    require valid-user
</Directory>
```

Si no está seguro de cómo crear los archivos de contraseña necesarios, remítase a capítulo anteriores.

Además, es posible restringir el acceso para un protocolo. Por ejemplo:

```
<Directory proxy:http:>
    . .
</Directory>
```

le permite controlar cómo se procesan las solicitudes HTTP en su servidor proxy. Del mismo modo, puede utilizar el esquema siguiente para controlar cómo maneja el servidor proxy cada protocolo:

```
<Directory proxy:ftp:>
    . .
</Directory>
```

```
<Directory proxy:https:>
. . .
</Directory>
```

También puede crear un host virtual exclusivamente para servidores proxy. En ese caso, las directivas deberían estar dentro del contenedor `<VirtualHost>`:

```
<VirtualHost proxy.host.com:>
. . .
</VirtualHost>
```

## **Escenario 3: crear una copia local de un sitio Web**

Un sitio Web mirror es una copia local de un sitio Web remoto. Puede utilizar el servidor proxy para realizar este tipo de copias del sitio Web `www.apache.org` para que sus usuarios puedan conectarse a ese sitio y acceder a la información de Apache rápidamente, puede utilizar el servidor proxy para crearlo, del siguiente modo:

```
ProxyPass / www.apache.org/
CacheRoot /www/cache
CacheDefaultExpire 24
```

Esto convierte el servidor proxy en un mirror del sitio Web `www.apache.org`. Esta configuración convierte a mi servidor proxy `blackhole.evoknow.com` en un mirror Apache. Cuando un usuario introduce la URL `http://blackhole.evoknow.com`, el usuario recibe la página de inicio del mirror Apache como si estuviese accediendo a `www.apache.org`.

Antes de que haga una copia local de un sitio Web, es importante que obtenga permiso, ya que suelen estar implicados asuntos relacionados con el copyright.

## **Preparar un navegador Web para utilizar un proxy**

Una vez que tiene configurado su servidor proxy, está preparado para establecer el navegador Web en sus host clientes. Los navegadores Web populares hacen muy sencilla la utilización de los servidores proxy. En las siguientes secciones, le mostraré cómo configurar Netscape Navigator 6 y Microsoft Internet Explorer (IE) 5.5 para proxy. Hay dos modos de establecer un servidor proxy para estos navegadores: configuración manual y configuración automática.

La configuración manual de los navegadores Web para proxy no es difícil. Sin embargo, si tiene que configurar muchos ordenadores de usuarios, este proceso se

puede convertir en un gran lío cada vez que tenga que cambiar su configuración proxy. Aquí es donde tiene sentido la configuración automática de proxy en los navegadores.

## Configuración manual del proxy

Se utiliza cuando tiene únicamente unas cuantas máquinas clientes y sus configuraciones de proxy no cambian con frecuencia. Si sus necesidades son distintas, por ejemplo, tiene cientos de máquinas clientes, debe saltarse esta sección y pasar a "Configuración automática del proxy."

### Configurar Netscape manualmente

Los siguientes pasos le guían en la configuración manual de Netscape:

1. Elija Editar>Preferencias del menú de Navigator. Debería ver una ventana de diálogo como la que se muestra en la figura 10.4.

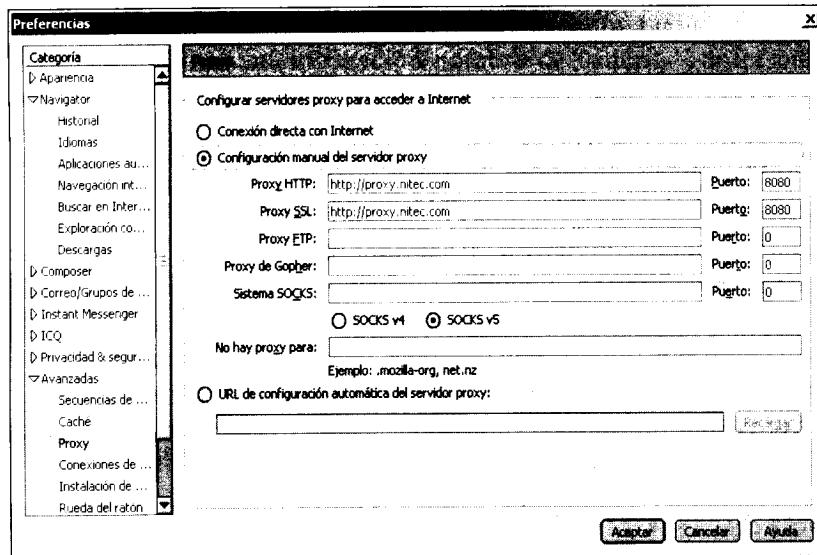


Figura 10.4. La ventana de establecimiento manual de un proxy en Netscape Navigator

2. Haga clic en la categoría Avanzadas.
3. Haga clic en la categoría Proxy.
4. Haga clic en la opción Configuración manual del servidor proxy.
5. Introduzca las URL del servidor proxy para los campos de entrada de datos HTTP, FTP y Security (HTTPS), junto con la información del puerto. Como estoy utilizando un solo servidor proxy para estos protocolos, la

URL (`http://proxy.evoknow.com /`) y el puerto (8080) son los mismos. Si tiene distintos servidores proxy para cada uno de estos servicios, tiene que especificarlos.

- Una vez que ha introducido la información, realice una solicitud de un documento remoto para saber si su proxy está funcionando. Un buen modo de determinar qué es lo que está ocurriendo es monitorizar el acceso al servidor proxy y los registros de error. En la mayoría de los sistemas Unix puede utilizar un comando como el que tiene a continuación para ver las entradas de registro tal y como se han escrito en el archivo:

```
tail -f /path/to/access/log
```

## Configurar Internet Explorer manualmente

Vamos a configurar Microsoft Internet Explorer para Windows. Para configurar manualmente Internet Explorer para proxy, siga los siguientes pasos:

- Elija Herramientas>Opciones de Internet. Esto abrirá la caja de diálogo que se muestra en la figura 10.5.

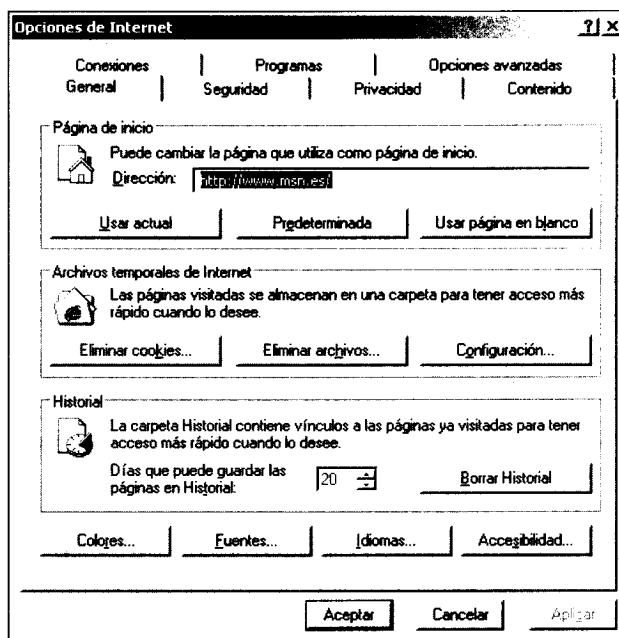
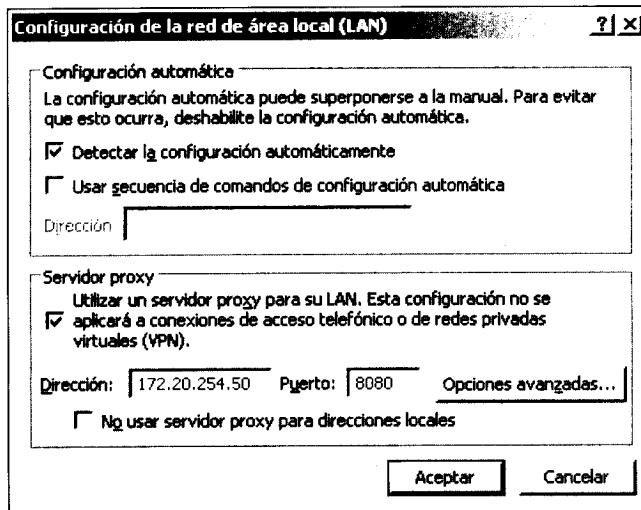


Figura 10.5. La ventana de establecimiento manual de un proxy en Microsoft Internet Explorer

- Haga clic en la pestaña de Conexiones y haga clic en el botón de Configuración de LAN. Supongo que está utilizando una conexión LAN gateway/router a Internet.

3. Seleccione la opción Utilizar un servidor proxy para su LAN...
4. Introduzca la URL del servidor proxy y el número de puerto en los campos proporcionados.
5. Haga clic en Aceptar para completar la configuración.

**TRUCO:** Si quiere especificar información sobre el servidor proxy distinta para los distintos protocolos, puede utilizar el botón Opciones avanzadas para obtener la ventana que se muestra en la figura 10.6.



**Figura 10.6.** La ventana de establecimiento avanzado de un proxy en Internet Explorer

En este caso, al igual que en el Netscape Navigator, puede especificar diferentes ajustes para los servidores proxy. Cuando haga clic en Aceptar y en el botón Aplicar para aplicar los nuevos ajustes, su navegador se configurará para utilizar el servidor proxy.

## Configuración automática del proxy

Los expertos en Netscape Communications pensaron en los problemas implicados en la configuración manual de un proxy para varios ordenadores clientes, y sugirieron un modo de superar este problema. Cuando se inicia el navegador Web, carga la función desde el archivo JavaScript (más tarde se discute cómo el navegador accede a este archivo) y llama a `FindProxyForURL` para cada solicitud URL. El navegador suministra los argumentos host y URL a la función para que pueda devolver la mayor parte de la configuración proxy.

**NOTA:** Microsoft también tiene opciones de autoconfiguración para Internet Explorer. Por desgracia, la incorporación de la autoconfiguración en el navegador en Microsoft es algo más complicada. Debe obtener el Internet Explorer Administrator Kit (IEAK) para crear archivos de autoconfiguración. Como obtener IEAK requiere un acuerdo de licencia que informe trimestralmente a Microsoft sobre la utilización de IEAK, este autor no ha obtenido el kit. Sin embargo, tengo la confirmación de una buena fuente de que la documentación IEAK tiene un escenario de configuración de proxy parecido al de Netscape y puede incluso utilizar los mismos scripts. Esta sección se aplica tanto a IE como a Navigator. La única diferencia es que si quiere utilizar IE, debe tener en cuenta la creación de archivos apropiados utilizando IEAK.

La autoconfiguración proxy se realiza utilizando un JavaScript especial. Esto es así tanto para Netscape Navigator como para IE. El JavaScript especial cumple estos requisitos:

- El JavaScript de autoconfiguración proxy debe implementar una función llamada `FindProxyForURL`. Esta función tiene el siguiente esquema:

```
function FindProxyForURL(url, host) {  
  
    // código java script  
    return "proxy to use for servicing the URL";  
}
```

- Los argumentos que recibe esta función son `url` y `host`. El argumento `url` es la URL completa que se ha solicitado y el argumento `host` es el nombre del host extraído de la URL. Por ejemplo, cuando el navegador Web detecta una solicitud para una página Web, llama a la función:

```
ret = FindProxyForURL("", )
```

**NOTA:** El argumento del host en la función es realmente una subcadena entre el `//` y el primer `:` o el primer `/`. El número de puerto no está incluido en este parámetro.

- La función debe devolver una cadena con la configuración proxy necesaria para una solicitud URL determinada. Los valores aceptables de las cadenas que representan una configuración proxy se muestran en la tabla 10.1.

**Tabla 10.1.** Valores de cadenas aceptables para la configuración del proxy

Cadena	Significado
NULL	Cuando se devuelve un valor NULL (no la cadena NULL), le dice al navegador que no utilice un proxy para esta solicitud.
DIRECT	Las conexiones se podrían realizar directamente, sin proxies.
PROXY host:port;	Debería utilizarse el proxy especificado.
SOCKS host:port;	Debería utilizarse el servidor SOCKS especificado.

## Asignar valores de retorno para FindProxyForURL

Tal y como se discute en la tabla 10.1, se pueden devolver cuatro valores. Obviamente, los valores interesantes son DIRECT y PROXY. Cuando tiene varios servidores proxy o servidores SOCKS, puede devolver una lista en lugar de un solo par host:port. Por ejemplo, la siguiente configuración proxy:

```
PROXY best-proxy.evoknow.com:8080; PROXY good-
proxy.evoknow.com:8081; PROXY soso-proxy.evoknow.com:8082
```

le dice al navegador que primero lo intente best-proxy.evoknow.com, y si falla, que lo intente el siguiente (good-proxy.evoknow.com), y así sucesivamente. Observe que cada par host:port está separado por un punto y coma y que la palabra clave PROXY se repite en cada par. Si fallan todos los servidores proxy, se le preguntará al usuario antes de intentar una conexión directa. Cuando fallan todos los proxies y no está especificada ninguna opción DIRECT, el navegador le pregunta al usuario si deberían ignorarse los proxies de forma temporal e intentar conexiones directas. Para evitar la interacción con el usuario la configuración anterior se convierte en la siguiente:

```
PROXY best-proxy.evoknow.com:8080; PROXY good-
proxy.evoknow.com:8081; PROXY soso-proxy.evoknow.com:8082;
DIRECT
```

Como la conexión directa ya está especificada como último recurso, no se le preguntará al usuario antes de realizar tal tipo de conexión en caso de fallo total del proxy. También puede mezclar PROXY y SOCKS. Por ejemplo:

```
PROXY best-proxy.evoknow.com:8080; SOCKS
socks4.evoknow.com:1080; DIRECT
```

En este caso, se utilizará el proxy basado en SOCKS, cuando el servidor proxy principal best-proxy.evoknow.com falle en la respuesta.

Cuando un proxy falla en la respuesta, el navegador Web vuelve a utilizar el proxy pasado 30 minutos. Si la subsolicitud posterior falla, el intervalo se alarga otros 30 minutos.

## Utilizar funciones predefinidas en FindProxyForURL

Para ayudar a los administradores Web (que además deben programar en JavaScript), hay disponible un conjunto de funciones predefinidas. En la tabla 10.2 se muestran estas funciones y su descripción.

**Tabla 10.2.** Funciones predefinidas para el script de configuración automática de programación de un proxy

Nombre de la función	Explicación	Ejemplos
isPlainHostName(host)	Devuelve true si no hay un punto en el nombre del host. En otras palabras, si no está incluido el nombre del dominio.	isPlainHostName("blackhole") devuelve true. isPlainHostName("blackhole.evoknow.com") devuelve false. evoknow.com.
DnsDomainIs(host, domain)	Devuelve true si el host pertenece al dominio. Observe que el nombre de dominio debe contener un punto inicial.	dnsDomainIs("www.evoknow.com", ".evoknow.com") devuelve true. dnsDomainIs("www.apache.org", ".evoknow.com") devuelve false. evoknow.com.
localhostOrDomainIs (host, fqdnhost)	Devuelve true si el host que forma parte de fqdnhost (fully qualified host name) coincide con el host.	localhostOrDomainIs ("a.b.com", "a.b.com") devuelve true. localhostOrDomainIs ("a.b", "a.b.com") devuelve true.  localhostOrDomainIs ("a.b.org", "a.c.com") devuelve false.
isResolvable(host)	Si un servidor DNS puede traducir el nombre del host a una dirección IP, devuelve true; en caso contrario, devuelve false. La utilización de esta función puede hacer más lentos los navegadores porque se necesita una consulta DNS para llevar a cabo la prueba.	isResolvable("{hyperlink}") devuelve true (porque {hyperlink} tiene registros DNS).

Nombre de la función	Explicación	Ejemplos
isInNet(host, IP address pattern, netmask)	Devuelve true si la dirección IP del host coincide con el patrón especificado en el segundo argumento. La búsqueda de coincidencia se realiza utilizando la máscara de red del siguiente modo: si uno de los octetos de la máscara es un 255, debe coincidir el mismo octeto de la dirección IP del host. Si un octeto de la máscara es 0, se ignora el mismo octeto de la dirección IP del host. La utilización de esta función puede hacer más lentos los navegadores porque se necesitará una consulta DNS para realizar la prueba.	Si el host tiene una dirección IP 206.171.50.51, isInNet(host, "206.171.50.50", "255.255.255.0") devuelve true porque, de acuerdo con la máscara de red, únicamente deben coincidir los tres primeros octetos y se debe ignorar el último.
dnsResolve(host)	Devuelve la dirección IP del host, en caso de éxito. Observe que la utilización de esta función puede hacer los navegadores más lentos porque es necesaria una consulta DNS para realizar la prueba.	dnsResolve("proxy.evoknow.com") devuelve "206.171.50.50".
myIpAddress()	Devuelve la dirección IP del host que está ejecutando el navegador Web. Observe que la utilización de esta función puede hacer los navegadores más lentos porque es necesaria una consulta DNS para realizar la prueba.	var hostIP = myIpAddress() devuelve el IP del host del navegador Web y almacena en él una variable llamada hostIP.
dnsDomainLevels (host)	Devuelve el número de niveles de dominio en el nombre del host.	dnsDomainLevels("www.nitec.com") devuelve 2.
shExpMatch(string, shellExpression)	Devuelve true si la cadena coincide con la expresión shell.	shExpMatch("path/to/dir", "*/*") devuelve true. shExpMatch("abcdef", "123") devuelve false.
WeekdayRange(weekday1, weekday2, gmt)	Únicamente es necesario el primer argumento weekday1. Devuelve true si el día en el que se ejecuta esta función es igual a weekday1 o si se encuentra en el rango de weekday1 a weekday2. Si el tercer parámetro, gmt, es GMT entonces se utiliza la hora GMT en lugar de la hora local. Los valores aceptables para weekday1 o weekday2 son SUN, MON, TUE, WED, THU, FRI o SAT.	weekdayRange("FRI") devuelve true si estamos a Friday (viernes) según la hora local. weekdayRange("MON", "FRI", "GMT") devuelve true si estamos en el rango de Monday-Friday (lunes a viernes) en la hora GMT.

Nombre de la función	Explicación	Ejemplos
dateRange(day)	Devuelve true si el día, el año y el mes actuales, o los tres, se encuentran en el rango. El valor de day (día) puede ser 1-31; el de month (mes) puede ser JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV o DEC; el valor de year (año) es un número de cuatro dígitos; gmt es "GMT" o nada (hora local).	dateRange(31) devuelve true si el día actual es el 31.
dateRange(day1, day2)		dateRange("JAN", "APR") devuelve true si el año actual se encuentra en el rango de January (enero) a April (abril).
dateRange(month)		
dateRange(month1, month2)		
dateRange(year)		
dateRange(year1, year2)		dateRange(1995) devuelve true si el año actual es 1995.
dateRange(day1, month1, day2, month2)		
dateRange(month1, year1, month2, year2)		
dateRange(day1, month1, year1, day2, month2, year2)		
dateRange(day1, month1, year1, day2, month2, year2, gmt)		
timeRange(hour)	Devuelve true si la hora, los minutos o los segundos especificados son los actuales. Si se especifica un rango, entonces	timeRange(9, 17) devuelve true si la hora actual se encuentra entre las 9 a.m. y las 5 p.m.
timeRange(hour1, hour2)		
timeRange(hour1, min1, hour2, min2)	devuelve true cuando la unidad de tiempo correspondiente se encuentra dentro del rango especificado. El valor de hour (hora) puede ser 0-23; el valor de min (minutos) puede ser 0-59; el valor de second (segundos) puede ser 0-59; y gmt es "GMT" o nada (hora local).	
timeRange(hour1, min1, sec1, hour2, min2, sec2)		
timeRange(hour1, min1, sec1, hour2, min2, sec2, gmt)		

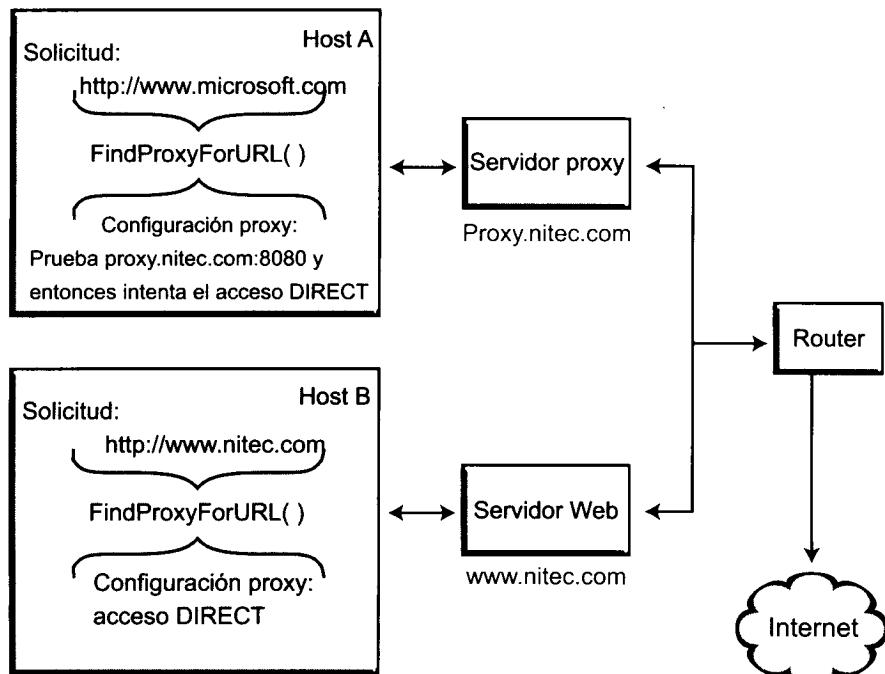
Con la ayuda de funciones predefinidas y con sus funciones personalizadas, puede escribir `FindProxyForURL` de modo que devuelva la cadena de configuración de proxy apropiada para cada solicitud.

A continuación proponemos unos cuantos ejemplos de escenarios en los que se puede escribir la función `FindProxyForURL` de distintos modos.

## Escenario 1: utilizar un proxy únicamente para solicitudes URL remotas

En este escenario, la idea es decirle al navegador Web que el proxy debería ser únicamente para solicitudes URL remotas, tal y como se muestra en la figura 10.7. En este caso, se evalúa una solicitud a `http://www.microsoft.com`, desde el host A, mediante el método `FindProxyForURL()`, el cual devuelve `proxy.nitec.com:8080` como el servidor proxy y entonces también le inscribe para que intente acceder directamente si se ha caído el proxy. De igual

modo, un navegador Web en el host B utiliza el método FindProxyForURL() para evaluar cómo acceder a <http://www.nitec.com>. La figura muestra que el método FindProxyForURL() instruye al navegador para que acceda al sitio directamente.



**Figura 10.7.** Utilizar un proxy únicamente para solicitudes URL remotas

El listado 10.1 es un ejemplo de una función FindProxyForURL.

**Listado 10.1.** Utilizar un proxy únicamente para solicitudes URL remotas

```

function FindProxyForURL(url, host) {
    // Comprueba si el host es un host local.
    // Si es un host local
    // especifica una conexión DIRECT (es decir, no proxy)
    // o utiliza el proxy.

    if (isPlainHostName(host) || dnsDomainIs(host, ".nitec.com"))

        return "DIRECT";

    else

        return "PROXY proxy.nitec.com:8081; DIRECT";
}

```

Cuando un usuario del servidor Web realiza una solicitud a URL `http://www.domain.com`, el navegador llama a `FindProxyForURL` con el argumento `url` asignado a `http://www.domain.com` y el host asignado a `www.domain.com`. La función llama primero a la función `isPlainHostName` para ver si la solicitud es sólo para el host (solo `www`) o no. Como no es así, `isPlainHostName` devuelve false. Ahora llama a la función `dnsDomainIs` para comprobar si se encuentra en el dominio `.nitech.com`. Esto también devuelve false. Como ambas pruebas devuelven false, se ejecuta la parte `else` de la sentencia condicional. En otras palabras, la solicitud URL de `http://www.domain.com` devuelve la siguiente configuración de proxy al navegador Web:

```
PROXY proxy.nitech.com:8081; DIRECT
```

Esto le dice al navegador Web que utilice el servidor proxy llamado `proxy.nitech.com` en el puerto 8081 si este no ha caído. Si ha caído, la solicitud debería servirse con una solicitud HTTP directa a `http://www.domain.com`. Para la mayor parte de las instalaciones proxy, esta configuración es suficiente. Vamos a ver un escenario más complicado.

## Escenario 2: utilizar varios servidores proxy

En este escenario, hay varios servidores proxy. La figura 10.8 nos muestra una red en la que hay tres servidores proxy: `http-proxy.nitech.com` que se utiliza para todas las solicitudes HTTP a las URL remotas; `ftp-proxy.nitech.com` que se utiliza para todas las solicitudes FTP a URL remotas; y `ssl-proxy.nitech.com` que se utiliza para todas las solicitudes HTTPS a URL remotas. El resto de las solicitudes URL que utilizan otros protocolos como GOPHER, NEWS, y similares, se conectan directamente. Todos los tipos de solicitudes locales se sirven también directamente. Para implementar esta configuración, `FindProxyForURL` se convierte en una función algo más compleja, algo parecido a lo que se muestra en el listado 10.2.

**Listado 10.2.** `FindProxyForURL` para una configuración de servidores multiproxy

```
function FindProxyForURL(url, host) {  
  
    //  
    // ¿Es esta la URL local? Si es así, utiliza una  
    // conexión DIRECT  
    //  
    if (isPlainHostName(host) ||  
        dnsDomainIs(host, ".nitech.com")) {  
  
        return "DIRECT";  
    } else {
```

```

    // De acuerdo, la URL es remota, por lo tanto, determina
    cuál es
    // proxy que hay que utilizar.

    if (url.substring(0, 5) == "http:") {

        return "PROXY http-proxy.nitec.com:8080";

    } else if (url.substring(0, 4) == "ftp:") {

        return "PROXY ftp-proxy.nitec.com:8080";

    } else if (url.substring(0, 6) == "https:") {

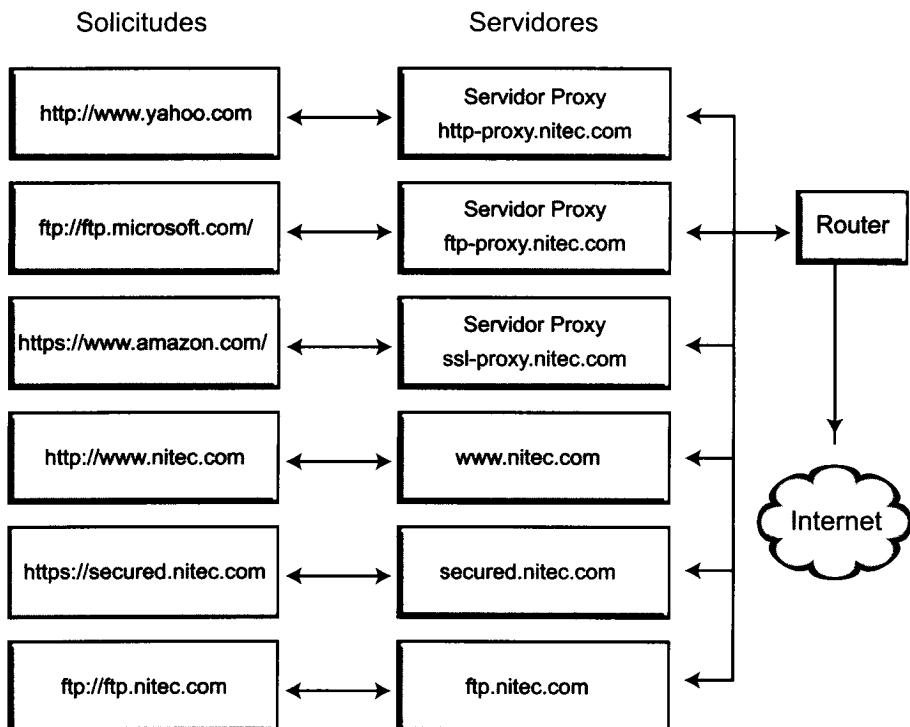
        return "PROXY ssl-proxy.nitec.com:8080";

    } else{

        return "DIRECT";

    }
}
}

```



**Figura 10.8.** Utilizar varios servidores proxy

Esta función comprueba primero si la solicitud URL es local. Si es local, entonces se sirve directamente. Si es una solicitud para un servidor remoto, se realiza una búsqueda de coincidencia del protocolo de la URL para localizar el servidor proxy adecuado. Sin embargo, sólo se reconocen los protocolos HTTP, FTP y HTTPS, y las URL que solicitan recursos remotos utilizando dichos protocolos son dirigidas a servidores proxy. Cuando una solicitud URL remota no coincide con ninguno de estos protocolos de estado, se conecta directamente.

## **Escenario 3: generar FindProxyForURL dinámicamente utilizando un script CGI**

También es posible personalizar la configuración de su servidor proxy basándose en el host al que está accediendo el servidor proxy. Esto se puede realizar utilizando un script CGI que realice una salida de `FindProxyForURL` de forma distinta dependiendo del `REMOTE_HOST` (el host del navegador). El listado 10.3 muestra uno de estos script, `proxy.pl`, escrito en Perl.

**Listado 10.3. proxy.pl**

```
#!/usr/bin/perl
#
# Un script Perl que da lugar a la configuración de un servidor
# proxy.
# $Author$
# $Revision$
# $Id$

# Obtiene la IP del host remoto de la variable de entorno CGI
# REMOTE_HOST
my $client = $ENV{REMOTE_HOST};

# Imprime el tipo de contenido necesario para permitir que el
# navegador sepa que se trata de una configuración proxy.
print "Content-type: application/x-ns-proxy-autoconfig\n\n";

# Si la solicitud procede de un host con la dirección IP
# 206.171.50.51 entonces saca la configuración del proxy
# desde una subrutina &specialClient
#
if ($client =~ /206\.171\.50\.51/){

    &specialClient;

} else {

    # Si la solicitud procede de cualquier otro cliente,
    entonces
        # envía la configuración proxy al resto de los clientes

    &otherClients;
```

```

}

exit 0;

sub specialClient{
#
# Esta subrutina saca una configuración del servidor proxy
#

print <<FUNC;

    function FindProxyForURL(url, host)
    {
        if (isPlainHostName(host) ||
            dnsDomainIs(host, ".nitech.com"))
            return "DIRECT";
        else if (shExpMatch(host, "*.com"))
            return "PROXY com-proxy.nitech.com:8080; "
        else if (shExpMatch(host, "*.edu"))
            return "PROXY edu-proxy.nitech.com:8080; "
        else
            return "DIRECT";
    }
FUNC
}

sub otherClients{
#
# Esta subrutina saca una configuración del servidor proxy
#

print <<FUNC;

    function FindProxyForURL(url, host)
    {
        return "DIRECT";
    }
FUNC
}

```

Este script da lugar a una configuración especial del servidor proxy para el host con la dirección IP 206.171.50.51; el resto de los host obtienen una configuración distinta. Para acceder a esta configuración proxy, puedo preparar el Netscape Navigator o el IE para dirigir este script a <http://www.nitech.com/cgi-bin/proxy.pl>. Por ejemplo, en IE puede especificar una URL igual a la anterior como la dirección del script de configuración

automática en Tools>Internet Options>Connections>LAN Settings>Use automatic configuration, excepto si le ha pedido al navegador que solicite un script CGI en lugar de un archivo .pac. Pero como el script envía el tipo de contenido de un archivo .pac, el navegador no tiene que preguntarse por qué obtiene la configuración proxy desde un script CGI en vez de desde un archivo .pac. Aunque el script del ejemplo no tiene muchas funciones, puede utilizar scripts parecidos para configuraciones proxy más complejas.



# 11 Ejecutar sitios Web perfectos

---

## En este capítulo

1. Creamos un ciclo Web para su organización.
2. Generamos sitios Web basados en plantillas utilizando el makepage.
3. Publicamos en una intranet utilizando el método PUT HTTP.
4. Estandarizamos sus estándares.
5. Hacemos sus Web más intuitivas.
6. Promocionamos su sitio Web en Internet.

En este momento, tendrá probablemente, uno o más sitios Web preparados y ejecutándose en su nuevo servidor Web Apache. Toda su empresa le está felicitando por su maravilloso trabajo. Está usted en el cielo, ¿verdad? ¡Incorrecto! Muy pronto sus compañeros le preguntarán cómo actualizar sus páginas en el sitio Web. Por ejemplo, el departamento de marketing podría llamar y preguntarle cómo actualizar la información referente a los precios, o el departamento legal le podría preguntar cómo puede añadir más contenido legal en uno de los sitios Web.

Esto es lo que les ocurre a los administradores Web de las organizaciones medianas y grandes. Estos administradores pronto se encuentran en el medio de un caos de solicitudes actualizadas y una lista de deseos. Por lo tanto, ¿cómo manejamos ahora la Web? En este capítulo, aprenderá a crear un entorno de gestión profesional de la Web que mantendrá a sus desarrolladores Web y a usted cuerdo y en sincronía con la Web.

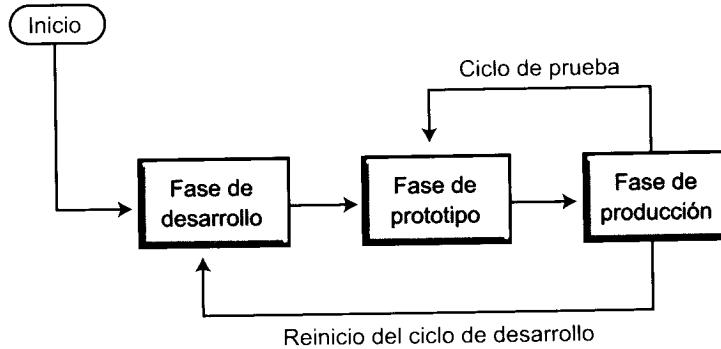
Este capítulo trata varios aspectos relacionados con el desarrollo de un sitio Web perfecto. Un sitio Web perfecto exhibe las características siguientes:

- Contenido de alta calidad: por supuesto, si no tiene un contenido útil y entretenido, no van a visitar su página. Sin embargo, el tipo de contenido que funciona depende del propósito de su sitio Web.
- Una presencia y un contenido consistentes: los sitios Web que tienen un tema consistente a lo largo de todas las páginas son más atractivos y suelen indicar un proceso elaborado. Crear una presencia y un contenido consistentes requiere herramientas y un proceso sistemático. Este capítulo le introduce en un proceso llamado *ciclo Web*, que requiere que utilice tres fases (desarrollo, prototipo y producción) para manejar sus sitios Web.
- Publicación automatizada: mi experiencia es que desarrollar constantemente contenido nuevo y excitante es un gran desafío en sí mismo. Si añade tareas manuales de presentación de contenido, muy pronto tendrá las cosas fuera de control. Por ejemplo, si tiene tres autores de contenido escribiendo páginas HTML para un sitio, debería plantear una presentación y un objetivo comunes, podría utilizar plantillas HTML e integrar el contenido utilizando un proceso automatizado. En este capítulo se discuten algunos de estos procesos.
- Prácticas estándar: para que el sitio Web sea intuitivo, hay una serie de guías que hay que seguir. Discutiremos algunas de las más importantes en este capítulo.

## Ciclo de desarrollo Web

Por desgracia, los típicos proyectos de desarrollo Web no comienzan con el diseño de un sitio Web manejable. En la mayoría de los proyectos, gran parte del tiempo se emplea en ejecutar y desarrollar el contenido; y no es habitual preocuparse sobre aspectos de gestión a largo plazo. Irónicamente, tan pronto como todo parece estar funcionando, las cosas empiezan a fallar debido a la falta de claridad, a la falta de un ciclo sostenible. En esta sección, trataremos el ciclo Web, el cual le permite crear una solución Web muy manejable.

Un ciclo Web consiste en tres fases: desarrollo, prototipo y producción. Implementando cada una de estas fases, puede crear un sitio Web manejable y sostenible. La figura 11.1 muestra un diagrama de alto nivel de un ciclo Web.



**Figura 11.1.** Un diagrama de alto nivel de un ciclo Web

Tal y como muestra la figura, un ciclo Web se inicia con la fase de desarrollo, continúa con la fase prototipo y termina con la fase de producción. Cuando se reinicia el ciclo, sin embargo, se inicia con la fase de producción y repite la ruta previa del ciclo. Las fases del ciclo son:

- **Fase de desarrollo:** en esta fase, comienza a desarrollar su contenido Web. El contenido, ya sea documentos HTML o scripts CGI o cualquier otra cosa, queda totalmente desarrollado y probado en esta fase. Una vez que los desarrolladores están totalmente seguros de que su trabajo está listo para integrarse con el sitio o sitios Web, el nuevo contenido desarrollado pasa a la siguiente fase.
- **Fase prototipo:** la fase de prototipo permite la integración del nuevo contenido desarrollado con el contenido existente, y permite el desarrollo de ciclos de prueba. Una vez en la fase de prototipo, los desarrolladores no siguen participando en el proceso. En este proceso, se introducen sujetos prueba que no son desarrolladores, para eliminar los posibles prejuicios a la hora de evaluar la corrección del contenido. En este momento, podrá localizar algunos problemas o quizás obtenga un conjunto satisfactorio de pruebas. En este último caso, está preparado para pasar el nuevo contenido, desarrollado, practicado y comprobado, a la fase de producción. Si se han encontrado problemas en este contenido nuevo, tendrá que comenzar desde la fase de desarrollo una vez que los desarrolladores hayan solucionado el problema en el área de desarrollo. No permita que los desarrolladores solucionen los problemas en el área prototipo.
- **Fase de producción:** esta fase consiste en la realización de una copia de seguridad (hacer un backup) y en las tareas de despliegue de contenido. Primero, ha de realizar una copia de seguridad de su contenido existente (funcional), y luego tiene que pasar el contenido de la fase de prototipo al espacio de producción Web. Esto tiene que suceder tan rápido como sea posible para reducir las desconexiones de visitantes y prevenir pérdidas de datos Web recolectados.

Cuando esté listo para empezar otro ciclo de desarrollo (reiniciar el proceso completo), copie el contenido de la fase de producción y páselo a la fase de desarrollo, de modo que los desarrolladores puedan trabajar en él. El ciclo continúa del mismo modo cada vez que lo necesite.

¿Qué ventajas tiene todo esto para usted? Le ofrece opciones de seguridad y de manejabilidad. Por ejemplo, si está desarrollando contenido y lo está descargando directamente en su sistema de producción antes de realizar la totalidad de las pruebas, está trabajando en condiciones límite. En la mayor parte de los casos, los desarrolladores de contenido querrán tener su contenido probado en el entorno local, y se darán prisa en terminar. Como un entorno de desarrollo local normalmente carece de integración de contenido actual con el contenido nuevo, las pruebas no son siempre realistas. Únicamente integrando el contenido existente con el nuevo, puede detectar las posibles incompatibilidades. Una carga directa en el sistema de producción desde la fase de desarrollo, sin la fase de prototipo, puede dar lugar a cualquiera de los siguientes errores:

- El nuevo contenido puede invalidar los archivos en el sistema de producción. Esto ocurre cuando tenemos archivos de imágenes, debido a la falta de una convención estándar a la hora de nombrar los archivos o debido a la utilización de directorios comunes para archivos de imagen.
- Los archivos de datos en el sistema de producción pueden ser invalidados, porque los desarrolladores de CGI utilicen los archivos antiguos de datos cuando desarrollan el contenido.
- Cuando hay implicados varios desarrolladores, algunos de los antiguos archivos pueden reaparecer en el servidor de producción, porque cada desarrollador podría empezar a trabajar con una copia en distintos momentos. Un desarrollador descarga su copia y luego lo hace el otro con el consiguiente error en el resultado.

Pueden aparecer muchos otros problemas si hay varios desarrolladores implicados y sus proyectos están interconectados. Si no quiere correr riesgos en la fase de producción, necesita pasar por la fase prototipo. Apache puede ayudarle a implementar estas fases.

## Poner en marcha el ciclo Web

Está preparado para poner en marcha el ciclo Web. De forma ideal, no deberíamos realizar ningún trabajo de desarrollo en el sistema del servidor de producción. Si su presupuesto no permite desplegar gran cantidad de máquinas para su Web, lo que si podría hacer es utilizar su servidor para implementar el ciclo.

Lo primero que necesita es preparar el servidor o los servidores para el ciclo Web. Aunque hay varios modos de hacerlo, vamos a discutir sólo tres. A continuación puede encontrar una breve descripción de cada uno de ellos.

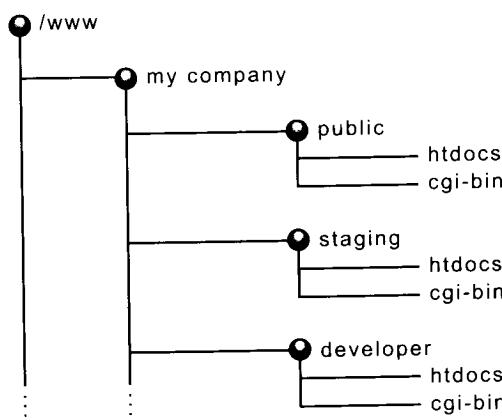
- **Un solo ordenador con dos hosts virtuales para desarrollo y prototipo.** El servidor de producción es el servidor principal de Apache. Tenga cuidado cuando modifique cualquier configuración de Apache en este sistema, ya que los cambios podrían afectar al comportamiento de los servidores de producción.
- **Un solo ordenador con tres servidores principales Apache para desarrollo, prototipo y producción.** Con este método, puede crear configuraciones separadas para cada servidor principal de Apache de modo que puede experimentar con las configuraciones Apache en el sitio de desarrollo sin perturbar la configuración de producción.
- **Al menos tres ordenadores distintos como servidores Apache de desarrollo, prototipo y producción.** Todos estos ordenadores ejecutan servidores Apache en el puerto 80.

## Establecer el ciclo Web

Puede establecer el ciclo Web de dos modos: puede utilizar dos host virtuales nuevos para implementar los sitios de desarrollo y de prototipo en su servidor de producción, o puede crear tres configuraciones Apache separadas para sus servidores de producción, de desarrollo y prototipo.

Si su trabajo de desarrollo incluye tocar los archivos de configuración Apache o probar un servidor Apache nuevo, debe utilizar archivos de configuración separados para el servidor de producción Apache y los otros dos servidores Apache. Si su desarrollo normal de Web no incluye cambios relacionados con Apache puede utilizar la aproximación de host virtual.

Un buen ciclo Web requiere una estructura de directorios perfectamente planeada. La figura 11.2 muestra esta estructura de directorios para un ciclo Web.



**Figura 11.2.** La estructura de directorios utilizada para los sitios public, stage y developer para "my company"

Esta figura muestra un buen ejemplo de una estructura de directorios porque le permite mantener los sitios público (public), de prototipo (staging) y de desarrollo (developer) bajo un solo directorio de máximo nivel (en este caso my company). Añadir un nuevo sitio Web significa crear una estructura de directorios parecida a ésta para él. En los ejemplos de configuración que vamos a discutir en las secciones siguientes, voy a suponer que tiene la estructura de directorios anteriores. También supongo que su host servidor Web se llama `www.mycompany.com`, y que tiene la dirección IP `206.171.50.50`. Asegúrese de que reemplaza estos valores con los adecuados para su propia configuración.

## Crear un host virtual para cada fase

Si ha decidido modificar los archivos de configuración de Apache como parte de su proceso de desarrollo, no utilice este esquema; necesita una sola asignación de archivos de configuración de Apache en este sistema, y cambiar los archivos para experimentación puede afectar a su servidor de producción. En ese caso, puede seguir utilizando una sola máquina, pero necesita ejecutar varios servidores Apache (principales). Esta aproximación se describe en la siguiente sección.

Si decide que no necesita realizar cambios relacionados con Apache en sus archivos de configuración, puede utilizar este esquema para crear un host virtual para cada fase. Para hacerlo, debería crear dos host virtuales que tengan el mismo nombre de servidor pero que se ejecuten en direcciones de puerto diferentes. La tabla 11.1 muestra una asignación de puerto para cada sistema.

**Tabla 11.1.** Asignaciones de puertos en servidores Apache para un ciclo Web

Puerto	Tipo de servidor
80	Servidor de producción (servidor principal).
1080	Servidor de prototipo (host virtual).
8080	Servidor de desarrollo (host virtual).

Puede elegir cualquier otra asignación de puertos, siempre que no utilice una dirección de puerto que ya se esté utilizando o que sea superior a 65535. El servidor de producción debería tener asignado el puerto 80, porque las solicitudes HTTP se envían por defecto a esta dirección de puerto. Para crear los host virtuales para la asignación de puerto mostrada en la tabla 11.1, necesita editar el archivo `httpd.conf` del servidor Apache del siguiente modo.

1. Para hacer que el servidor Apache escuche estos puertos, utilice la directiva `Listen`:

```
Listen 80
Listen 1080
Listen 8080
```

## 2. Entonces, puede crear dos host virtuales del siguiente modo:

```
# No olvide cambiar la dirección IP, los valores de las
directivas
# ServerName, DocumentRoot, ScriptAlias,
# TransferLog y ErrorLog con los valores apropiados en su
# caso, para su propio sistema de configuración.
#
<VirtualHost 206.171.50.50:1080>
    ServerName www.mycompany.com
    DocumentRoot "/www/mycompany/staging/htdocs"
    ScriptAlias /cgi-bin/ "/www/mycompany/staging/cgi-bin/"
    TransferLog logs/staging-server.access.log
    ErrorLog logs/staging-server.error.log
</VirtualHost>

<VirtualHost 206.171.50.50:8080>
    ServerName www.mycompany.com
    DocumentRoot "/www/mycompany/developer/htdocs"
    ScriptAlias /cgi-bin/ "/www/mycompany/developer/cgi-
bin/"
    TransferLog logs/developer-server.access.log
    ErrorLog logs/developer-server.error.log
</VirtualHost>
```

**NOTA:** En el ejemplo anterior, se utiliza la misma dirección IP en ambos host, pero se especifican puertos distintos en el contenedor <VirtualHost ...>. La dirección IP es la misma que la del servidor principal, www.mycompany.com. La directiva ServerName también está asignada al nombre del servidor principal. La configuración de su servidor principal será la habitual.

La URL `http://www.mycompany.com:1080` puede utilizarse para acceder al sitio prototipo; para acceder al sitio de desarrollo, esta URL puede utilizar: `http://www.mycompany.com:8080`.

## Utilizar varios procesos (principales) del servidor Apache

Podría utilizar más de un proceso de un servidor (principal) si ha decidido experimentar con Apache fuera de la fase de desarrollo. Tiene que crear tres conjuntos de archivos de configuración, con cada conjunto dirigido a un DocumentRoot y un ScriptAlias distintos. Una vez que lo ha hecho, puede arrancar los tres servidores (principales) Apache desde el directorio principal en el que tenga instalado Apache (ejemplo: /usr/local/apache) del siguiente modo:

```
httpd -f conf/httpd.conf
httpd -f conf/staging/httpd.conf
httpd -f conf/developer/httpd.conf
```

Cuando decida compilar una nueva versión de Apache y la ejecute bajo el servidor de desarrollo, puede simplemente cebarlo en el archivo de configuración para el servidor de desarrollo. Por ejemplo, si ha decidido añadir un nuevo módulo y quiere ver el efecto de este módulo en su contenido, puede simplemente ejecutar los servidores de desarrollo y de prototipo utilizando ese ejecutable en lugar del ejecutable de su servidor de producción (`httpd`). Una vez que ha compilado el nuevo ejecutable tendrá que volver a nombrarlo con algo parecido a `httpd-xx80`, para asegurar que no invalida accidentalmente el ejecutable del servidor de producción.

Para implementar el ciclo, siga las siguientes instrucciones.

1. Ha de crear dos subdirectorios en el directorio de configuración de Apache llamados `staging` y `developer`, del siguiente modo:

```
mkdir /path/to/Apache/server/root/conf/staging  
mkdir /path/to/Apache/server/root/conf/developer
```

**NOTA: No olvide reemplazar `/path/to/Apache/server/root/conf` con la ruta actual del directorio de configuración de su servidor.**

2. Copie todos los archivos `*.conf` en los subdirectorios `staging` y `developer`, del siguiente modo:

```
cp /path/to/Apache/server/root/conf/*.conf /path/to/Apache/  
server/root/conf/staging/*  
cp /path/to/Apache/server/root/conf/*.conf /path/to/Apache/  
server/root/conf/developer/*
```

3. Modifique el archivo `httpd.conf` en el subdirectorio `staging` para que escuche el puerto 1080 en lugar del puerto 80, que es el puerto por defecto. Puede utilizar la directiva `Port` o la directiva `Listen` para realizarlo. Del mismo modo, tiene que modificar `httpd.conf` en el subdirectorio `developer` para que la directiva `Port` y la directiva `Listen` tengan asignado el puerto 8080.
4. Modifique los archivos `srm.conf` (o los `httpd.conf`) en los subdirectorios `staging` y `developer` para que dirijan sus directivas `DocumentRoot` y `ScriptAlias` a la ruta adecuada. Los cambios necesarios aquí para la estructura de directorios mostrada en la figura 11.2 son:

```
DocumentRoot      "/www/mycompany/staging/htdocs"  
ScriptAlias       /cgi-bin/    "/www/mycompany/staging /cgi-bin/"
```

para la configuración del sitio prototipo. Por otro lado, para el archivo de configuración del sitio de desarrollo, serán:

```
DocumentRoot      "/www/mycompany/developer/htdocs"
ScriptAlias      /cgi-bin/    "/www/mycompany/developer/cgi-bin/"
```

**NOTA:** Si utiliza una configuración especial para su servidor de producción, el cual utiliza información de la ruta absoluta, tendrá que editar los nuevos archivos de configuración, para los subdirectorios `staging` y `developer`.

## Utilizar varios ordenadores servidores Apache para el ciclo Web

Si puede tener varios ordenadores servidores de Apache (es decir, uno para desarrollo, uno para prototipo y uno para producción) para crear el entorno del ciclo Web, no necesita crear configuraciones Apache especiales. Simplemente instale Apache en todos los host implicados, y trate un host como el host del sitio de desarrollo, un segundo host como el host del sitio de prototipo y el tercer host como el sitio de producción. Como ahora tiene servidores Apache ejecutándose en tres host distintos, puede ejecutar también cada servidor en el puerto 80. Eso es todo lo que necesita para un ciclo con varios host.

## Implementar el ciclo Web

Para iniciar su ciclo Web, copie el contenido de producción de su directorio raíz de documentos del servidor de producción en su sitio de desarrollo. Por ejemplo, si su configuración es una de las dos primeras de la lista anterior, puede copiar fácilmente todo su contenido Web en el sitio de desarrollo utilizando los siguientes comandos Unix:

```
cd  /path/to/production/docroot/dir
tar cvf - . . | (cd /path/to/development/site/docroot/dir ;
tar xvf - )
```

Estos comandos copian todos los archivos y directorios de su servidor de producción en la raíz de documentos de su servidor de desarrollo. Simplemente tiene que asegurarse de cambiar la información de la ruta donde sea necesario en su sistema.

Para un entorno con varios ordenadores para el ciclo Web, puede crear un archivo `tar` en su servidor de producción y copiarlo en su sitio de desarrollo vía FTP.

Asigne los permisos de archivos de modo que Apache pueda leer todos los archivos y ejecutar scripts CGI. Si tiene directorios en los que Apache podría tener acceso de escritura (para scripts CGI que escriban datos), debería asignar también estos permisos. Una vez que lo ha hecho, inicie o reinicie Apache para servir el sitio de desarrollo.

Ahora, tiene que asegurarse de que aparece el sitio de desarrollo (en el navegador Web) igual que el sitio de producción. Realice algunas comparaciones manuales y algunas pruebas. Asegúrese de que los script también funcionan.

Si cualquiera de sus scripts CGI produce las URL codificadas por hardware para su servidor de producción, seguirán haciendo lo mismo para su sitio de desarrollo. Puede ignorar estas URL o dejarlas fijas de modo que utilicen la variable de entorno SERVER\_NAME y la dirección de puerto de SERVER\_PORT.

## Probar el ciclo Web

Cuando todo funciona como debe funcionar, significa que ha creado con éxito un entorno de ciclo Web. Ahora puede pedirle a sus desarrolladores que pongan nuevo contenido y scripts en el sitio de desarrollo y que lo prueben. Cada vez que se completa un desarrollo de contenido, debe probarlo primero en el área de desarrollo. Las pruebas suelen enfocarse en los siguientes eventos:

- ¿Cumple su propósito? En otras palabras, ¿la funcionalidad que proporciona el nuevo contenido cumple sus especificaciones?
- ¿Tiene efectos secundarios el nuevo contenido? Por ejemplo, si el contenido nuevo es realmente un nuevo script CGI, debería utilizar el soporte de depuración de errores de Apache para controlar cómo funciona el script.

## Mover el sitio nuevo al servidor de producción

Una vez que está satisfecho con los resultados de las pruebas, evite tener que realizar otro conjunto de pruebas de funcionalidad suspendiendo cualquier desarrollo más allá de su nuevo contenido. Cuando llegue el momento de actualizar un sitio de producción, realice una copia de su sitio de producción y colóquela en el sitio prototipo. A continuación tenemos algunos trucos para hacerlo:

- Asegúrese de que el sitio prototipo es exactamente el mismo que el sitio de producción. Una vez que ha realizado algunas comprobaciones manuales para asegurarse de que todo tiene la misma presentación y el mismo enfoque, puede mover contenido y scripts nuevos al sitio prototipo e integrarlos.
- Mueva un proyecto cada vez, de modo que pueda encontrar y resolver los problemas de prototipo. Por ejemplo, si añade tres scripts CGI nuevos a su sistema, mueva un script cada vez al área prototipo. Realice tanto las pruebas de funcionalidad como las de integración del sitio. Si el script supera la prueba, mueva el siguiente script al área prototipo. Una vez que ha movido todo el contenido nuevo, puede realizar las pruebas de nivel de integración del sitio. Monitorice con cuidado los registros del sitio de prototipo. ¿Nota algo raro en los registros de error? Si no es así, entonces está preparado para realizar una actualización de su sitio de producción. Ha de tener cuidado al llevar esto a cabo. Por ejemplo, si tiene algún script CGI

en el servidor de producción que crea archivos de datos en el área de producción, no debería invalidar ninguno de estos archivos de datos con lo que tiene en el área prototípico.

- El mejor momento para actualizar su sitio de producción es cuando piensa que el servidor de producción está menos ocupado. En este momento, puede coger los archivos de datos de su servidor de producción y colocarlos en los directorios apropiados en la versión prototípica del sitio. Todo esto deja a su sitio prototípico en sincronía con el sitio de producción. En este momento, tiene que descargar rápidamente su sitio prototípico dentro del área de producción. Esto podría ser muy delicado porque el sitio de producción está vivo, y nunca va a saber cuándo podría acceder un visitante a una página o cuándo va a utilizar un script CGI que necesita para leer o escribir archivos de datos.

Para minimizar el tiempo (al menos en un sistema de un solo servidor) puede crear un script shell que haga lo siguiente:

1. Copie todos los archivos de datos en las áreas apropiadas de su sitio prototípico.
2. Vuelva a nombrar el directorio de producción de máximo nivel (como el directorio `public` de la figura 11.2) a algo del tipo `public.old`.
3. Vuelva a nombrar su directorio prototípico de máximo nivel (como el directorio `staging` de la figura 11.2) como acostumbre llamar a su directorio de producción de máximo nivel, por ejemplo, `public`.
4. Vuelva a nombrar el viejo directorio de producción (como `public.old`) como acostumbre llamar a su directorio prototípico de máximo nivel, por ejemplo, `staging`.

De este modo, el sitio prototípico se convierte en el sitio de producción en sólo unos cuantos pasos, sin necesidad de realizar un gran número de operaciones de copias de archivos. En la lista 11.1 tenemos un ejemplo de un script que corresponde al entorno mostrado en la figura 11.2.

#### Listado 11.1. script stage2production.sh

```
#!/bin/sh
# Propósito: un simple script shell para copiar archivos de datos
# en el área prototípico y para volver a nombrar el área prototípico
# en un sitio vivo de producción. También vuelve a nombrar el
# área antigua de producción en el área prototípico.
#
# Copyright © 2001 Mohammed J. Kabir
# License: GNU Public License
#
# Necesitará cambiar estas variables para utilizar este script.
```

```

DATA_FILES="/www/mycompany/public/htdocs/cgi-data/*.dat";
TEMP_DIR="/www/mycompany/public.old";
PRODUCTION_DIR="/www/mycompany/public";
STAGE_DIR="/www/mycompany/staging";

# Copie los datos en el directorio staging.
/bin/cp $DATA_FILES $STAGE_DIR

# Nombre de forma temporal el directorio de producción actual
TEMP_DIR
/bin/mv PRODUCTION_DIR TEMP_DIR

# Nombre el sitio prototipo actual como el directorio de
# producción
/bin/mv STAGE_DIR PRODUCTION_DIR

# Nombre de forma temporal el directorio (antiguo) de producción
# como el directorio staging
/bin/mv TEMP_DIR STAGE_SITE

# Para estar seguros, cambie los permisos asignados del directorio
# de producción para que el usuario Apache (httpd)
# y el grupo Apache (httpd) puedan leer todos los archivos.
# Si utiliza algún otro usuario y grupo para Apache, tiene
# que modificar este comando de acuerdo con su sistema.

/bin/chown -R httpd.httpd $PRODUCTION_DIR

# Cambie el permiso de archivo de modo que el dueño
# (httpd en este caso) tenga permiso de lectura, escritura y de
# ejecución, el grupo (httpd en este caso)
# tenga permiso de lectura y de ejecución, y el resto tenga
# permiso para ver los archivos de
# producción

/bin/chmod - R 750 $PRODUCTION_DIR

```

Una vez que ejecutamos este script, debería realizar una prueba rápida para asegurarse de que todo está en orden. En caso de problemas, puede volver a nombrar el directorio de producción actual, y cambiar el nombre del directorio prototipo al nombre de su directorio de producción para restablecer su último sitio de producción.

## Construir un sitio Web utilizando plantillas y el makepage

Mantener un ciclo Web estricto le proporciona un proceso que se puede repetir para la publicación de grandes sitios Web. Sin embargo, sigue necesitando un

proceso de presentación de contenido que está altamente automatizado y que requiera muy poca participación por nuestra parte.

Existe una gran cantidad de programas en el mercado que nos pueden ayudar en este proceso.

Algunos desarrolladores utilizan Microsoft Front Page para gestionar el desarrollo de contenido; otros utilizan el Dreamweaver; y algunos utilizan otros productos. Algunas personas utilizan métodos más robustos de compañías de desarrollo Web que cuestan cientos o miles de dólares. A continuación tenemos una solución que me ha funcionado (y lo hará durante años) en el mantenimiento de sitios Web. Los requisitos para esta solución son:

- Crear un mecanismo sencillo para que los autores de contenido puedan publicar páginas Web con una presentación y un objetivo consistentes.
- Requiere una cantidad de trabajo mínima para el autor de contenido de modo que la mayor parte del trabajo está automatizado.
- Supone que el desarrollador de contenido sabe muy poco HTML y prefiere enviar el contenido en formato de texto.

Para implementar esta solución, he escrito el `makepage`, un script que está incluido en el CD-ROM de este libro. Este script utiliza un conjunto de plantillas HTML y una página con el texto del cuerpo (contenido) para construir cada página en el sitio Web. Cuando empecé este proyecto, quería generar cada página al vuelo utilizando CGI o `mod_perl`, pero más tarde decidí generar el contenido una vez al día porque mis sitios iban a ser actualizados únicamente una o dos veces al día. Sin embargo, es muy sencillo aumentar la frecuencia de actualización, como aprenderá en esta sección. El script `makepage` supone que cada página consiste en:

- Una barra de navegación a la izquierda.
- Una barra de navegación a la derecha.
- Un menú de navegación central.
- Un área que aloja el contenido de la página.

Cada vez que se ejecuta el script `makepage` en un directorio determinado, busca todos los archivos terminados con la extensión `.txt` y crea las páginas `.html` correspondientes. Por ejemplo, si ejecuta el script `makepage` en un directorio con un archivo de texto llamado `index.txt`, el script se ejecutará y producirá una salida parecida a esta:

```
Processing ./index.txt
RSB template ./-rsb.html chosen: /home/mjkabir/www/default-
rsb.html
LSB template ./-lsb.html chosen: /home/mjkabir/www/default-
lsb.html
```

```
Backed up ./index.html as ./index.html.bak
    Template: /home/mjkabir/www/default-tmpl.html
    BODY file: ./index.txt
    LSB file: /home/mjkabir/www/default-lsb.html
    RSB file: /home/mjkabir/www/default-rsb.html
Bottom Nav file: /home/mjkabir/www/default-bottom.html
Top Nav file: /home/mjkabir/www/default-top.html
    HTML file: ./index.html
```

Esta salida del script muestra que se está procesando el archivo `index.txt` en el directorio actual (indicado por el punto). Entonces muestra cuál es el archivo de la plantilla de la barra de navegación de la derecha, RSB (Right Side Navigation Bar), que se está utilizando para `index.html`. El script busca primero el archivo de plantilla RSB llamado `index-rsb.html` y si no encuentra un archivo RSB específico para el archivo de texto, utiliza el archivo RSB por defecto para el directorio completo, que va a ser el `default-rsb.html`. Repite el mismo proceso para la plantilla de la barra de navegación de la izquierda, LSB (Left Side Navigation Bar), seleccionada.

Entonces realiza una copia del actual `index.html` (la salida de la última ejecución) a `index.html.bak` y utiliza la plantilla con el cuerpo por defecto `default-tmpl.html` para crear la página `index.html`. Si encuentra una plantilla de cuerpo llamada `index tmpl.html`, utiliza esta plantilla en lugar de la plantilla de cuerpo por defecto del directorio.

Esto le ofrece la flexibilidad en el diseño de cada página Web. Puede crear simplemente un gran directorio de plantillas y tener todas las páginas con el mismo aspecto.

O puede personalizar una sola página en el directorio con sus propias plantillas RSB, LSB y BODY.

Si ejecuta el script en el directorio raíz de documentos utilizando el comando `makepage path_to_document_root`, el script crea páginas automáticamente en todos los subdirectorios en la raíz de documentos. De este modo, puede establecer este script como un trabajo del `cron` que se ha de ejecutar cada hora, o a diario, o a la semana, o incluso cada minuto según sus necesidades de actualización.

Los autores de contenido simplemente lanzan sus archivos de texto y las páginas se crean automáticamente.

Cuando se lanza un nuevo archivo de texto y no se proporciona la plantilla RSB, LSB o BODY específica para la página, ésta se crea con la plantilla por defecto del directorio, lo que facilita enormemente el que se añadan nuevas páginas. Simplemente escriba una página en su editor de texto favorito y FTP el archivo en su directorio correcto de su sitio Web y éste será publicado en el siguiente `makepage` ejecutado vía `cron`.

El paquete `makepage` suministrado en el CD-ROM incluye las plantillas por defecto que puede estudiar para construir las suyas propias.

# Utilizar HTTP PUT para publicaciones Web en una Intranet

Apache soporta el método PUT, que le permite publicar una página Web. Sin embargo, esta característica tiene asociados grandes riesgos de seguridad en el caso de que no se implemente con cuidado extremo. Es evidente que no desea que nadie pueda cambiar su sitio Web. Únicamente recomiendo utilizar esta característica para intranets que no son accesibles desde Internet.

Necesita el módulo `mod_put`, que implementa los métodos PUT y DELETE encontrados en HTTP 1.1. El método PUT le permite cargar contenidos en el servidor y el método DELETE le permite eliminar recursos del servidor. Puede bajar este módulo de [http://hpwww.ec-lyon.fr/~vincent/apache/mod\\_put.html](http://hpwww.ec-lyon.fr/~vincent/apache/mod_put.html).

## Las directivas del módulo `mod_put`

El módulo `mod_put` proporciona tres directivas para controlar la publicación basada en PUT y en DELETE.

### **EnablePut**

`EnablePut` activa o desactiva el método PUT. Para utilizar el método PUT, debe activarlo asignándole el valor On a esta directiva.

**Sintaxis:** `EnablePut On|Off`

**Predefinido:** `EnablePut Off`

**Contexto:** directorio, localización

### **EnableDelete**

`EnableDelete On | Off` activa o desactiva el método DELETE, que le permite eliminar una página Web vía HTTP. Para utilizar el método DELETE, debe activarlo asignando el valor On a esta directiva.

**Sintaxis:** `EnableDelete On | Off`

**Predefinido:** `EnableDelete Off`

**Contexto:** directorio, localización

### **umask**

`umask octal_` determina la máscara de permiso por defecto (es decir, `umask`) para un directorio. El valor por defecto de 007 asegura que cada archivo dentro del directorio se crea con permiso 770, que únicamente permite al dueño del archivo y al grupo leer, escribir y ejecutar el archivo.

**Sintaxis:** umask octal\_value

**Predefinido:** umask octal\_007

**Contexto:** directorio, localización

## Compilar e instalar mod\_put

Una vez que ha bajado mod\_put, tiene que realizar los siguientes pasos para compilarlo e instalarlo:

1. Extraiga la fuente mod\_put.tar.gz y mueva el directorio que ha creado dentro del subdirectorio de módulos de su distribución fuente de Apache.
2. Añada el módulo mod\_put a Apache utilizando el script configure (o config.status si ya lo tiene compilado en Apache). Ejecute el script con la opción add --enable-module=put.
3. Compile e instale Apache utilizando el comando make && make install.
4. Reinicie Apache utilizando el comando /usr/local/apache/bin/apachectl.

## Establecer un directorio Web que permita el método PUT

Los clientes Web como Netscape, AOLPress y Amaya, pueden publicar páginas Web mediante el método PUT. Esta sección le enseña cómo establecer httpd.conf para permitir publicaciones basadas en PUT para un solo directorio Web bajo su árbol raíz de documentos.

**ADVERTENCIA:** Tenga cuidado con el método PUT si tiene pensado utilizarlo fuera de su intranet. Utilizar PUT en un sitio Web accesible al resto del mundo en Internet podría aumentar enormemente los riesgos de seguridad porque alguien puede desfigurar su sitio Web si el proceso de autentificación basado en la Web que se ha descrito aquí, se ve comprometido. Sólo recomiendo utilizar el método PUT para uso interno.

1. Tiene que crear la configuración siguiente en httpd.conf:

```
Alias location_alias  
"physical_directory_under_document_root"  
  
<Location location_alias>
```

```

    EnablePut On
    AuthType Basic
    AuthName "Name_of_the_Web_section"
    AuthUserFile path_to_user_password_file
    <Limit PUT>
        require valid-user
    </Limit>

</Location>

```

A continuación vamos a describir lo que está ocurriendo en el código anterior (para aprender más sobre estas directivas relacionadas con la autenticación, lea el capítulo 7):

- Un alias llamado `loc_alias` se asocia con una ruta física llamada `physical_directory_under_document_root`.
- El directorio `<Location>` asigna directivas para este alias.
- La directiva `EnablePut` activa el módulo `mod_put`.
- La directiva `AuthType` asigna el tipo de autenticación en autenticación HTTP Basic.
- La directiva `AuthName` asigna una etiqueta para esta sección. Esta etiqueta se despliega en la caja de diálogo de autenticación que los navegadores Web muestran al usuario, por lo que debe asegurarse de que es significativa.
- `AuthUserFile` determina el archivo de contraseña de usuario que utiliza para autenticar al usuario.
- El contenedor `<Limit>` fija límites para el método PUT. Le dice a Apache que necesita usuarios válidos cuando un cliente Web envía una solicitud PUT.

A continuación se muestra un ejemplo de la configuración definida anteriormente:

```

Alias /publish/ "/www/mysite/htdocs/publish/"

<Location /publish>
    EnablePut On
    AuthType Basic
    AuthName "Web Publishing Section"
    AuthUserFile /www/mysite/secrets/.users

    <Limit PUT>
        require valid-user
    </Limit>

</Location>

```

En este ejemplo, el directorio físico /www/mysite/htdocs/publish tiene activado un método PUT para todos los usuarios del archivo /www/mysite/secrets/.users.

2. Reinicie el servidor Apache utilizando el comando /usr/local/apache/bin/apachectl restart y utilice su navegador Web con soporte PUT para publicar un documento en el directorio http://your\_web\_server/loc\_alias. Para la configuración del ejemplo, esta URL es http://server/publish.

Cuando se publica un archivo utilizando el método PUT, tendrá el permiso asignado utilizando la directiva umask para el módulo mod\_put. El archivo pertenecerá al usuario bajo el que se está ejecutando Apache. Por ejemplo, si asigna las directivas User y Group en httpd.conf para que sean httpd, entonces el archivo es propiedad del usuario httpd, y el propietario del grupo es también dueño de httpd.

## Establecer un host virtual para utilizar el módulo mod\_put

El usuario determina en la directiva User en httpd.conf archivos propios creados por mod\_put. Esto supone un problema en los sitios con varios usuarios distintos porque ahora todo el mundo puede invalidar los archivos de otra persona utilizando el método PUT. Puede resolver este problema fácilmente utilizando un host virtual para cada usuario, tal y como se muestra a continuación.

1. Añada las líneas siguientes a httpd.conf:

```
ChildPerUserID number_of_chid_servers username1 groupname1
```

en el que el par username1 groupname1 son el usuario y el grupo que hay que utilizar para un host virtual. Cambie estos nombres con los nombres de usuario y de host que está utilizando. Ha de crear tantas líneas ChildPerUserID como necesite. num\_of\_chid\_servers es un número que utiliza Apache para lanzar procesos hijo asociados con este host virtual. Por ejemplo, si tiene dos usuarios llamados carol y john y quiere asignar 10 hijos Apache por cada host virtual, entonces añada las siguientes líneas en httpd.conf:

```
ChildPerUserID 10 carol carol_group  
ChildPerUserID 10 john john_group
```

Asegúrese de que los usuarios y grupos existen realmente en /etc/passwd y en /etc/group, respectivamente.

2. Ha de crear un VirtualHost para cada usuario que necesite publicación PUT del siguiente modo:

```

NameVirtualHost IP_Address

<VirtualHost IP_Address>

    ServerName vhost_domain_name
    AssignUserID user_name group_name

    Alias location_alias
    "physical_directory_under_document_root"

    <Location location_alias>
        EnablePut On
        AuthType Basic
        AuthName "Name_of_the_Web_section"
        AuthUserFile path_to_user_password_file

        <Limit PUT>
            require username
        </Limit>

    </Location>

    # Otras directivas

</VirtualHost>
Example:
NameVirtualHost 192.168.1.100

<VirtualHost 192.168.1.100>
    ServerName carol.domain.com
    AssignUserID carol carol_group
    DocumentRoot /www/intranet/htdocs/carol

    Alias /publish/ "/www/intranet/htdocs/carol/publish/"

    <Location /publish>
        EnablePut On
        AuthType Basic
        AuthName "Carol's Publishing Site"
        AuthUserFile /www/intranet/secrets/.users

        <Limit PUT>
            require carol
        </Limit>

    </Location>

</VirtualHost>

```

Se puede publicar el usuario carol en el directorio `http://carol.domain.com/publish` utilizando su propia cuenta de usuario. También podrá acceder a los archivos creados por el servidor vía FTP, ya que los archivos son propiedad del usuario carol.

3. Una vez que ha creado un host virtual para cada usuario, reinicie Apache utilizando el comando /usr/local/apache/bin/apachectl restart y probando el sistema de cada usuario publicando una página de prueba utilizando la URL apropiada.

## Mantenimiento de su sitio Web

Una vez que ha implementado el ciclo Web y que tiene un proceso localizado de generación de contenido, es importante mantener su Web. Las tareas habituales de mantenimiento incluyen la monitorización del servidor, y el registro y la copia de seguridad de los datos. Los aspectos de monitorización y registro en el servidor se discuten en el tema 8. Esta sección trata la realización de copias de seguridad de los datos. Debería tener, si es posible, dos tipos de backup, backup online y backup offline.

### Backup online

El backup online es útil en caso de emergencia. Puede acceder rápidamente a los datos copiados y, en la mayor parte de los casos, realizar en pocos minutos las tareas necesarias de restitución. Para obtener una solución backup online, buscar un fabricante de backup online comercial o hablar con su ISP. Si está alojando su servidor o servidores Web en su propia red, sin embargo, puede mantener backups de otro host en su red. En la mayoría de los sistemas Unix, puede ejecutar un programa llamado `rdist` para crear directorios mirror en sus sitios Web en otros host Unix (en el capítulo 23 tiene un ejemplo de una aplicación de mirror de un sitio basado en `rdist`).

Podría ser una buena idea mantener una versión comprimida de los datos Web en el servidor en sí. En los sistemas Unix, puede establecer un trabajo `cron` para crear un archivo `tar` comprimido con los datos de la Web con una frecuencia determinada. Por ejemplo:

```
# Para sistemas V-ish Unix, rango laborable 0?6 donde 0= Domingo
# Para sistemas BSD-ish utiliza rango laborable 1?7 donde
1=Lunes
# Es un ejemplo es para un sistema Linux (System V-ish cornd)
30 2 * * 0,1, 3, 5,    root   /bin/tar czf /backup/M-W-F-Sun.tgz
/www/*
30 2 * * 2, 4, 6      root   /bin/tar czf /backup/T-TH-Sat.tgz  /
www/*
```

Si estas dos entradas de `cron` se guardan en `/etc/crontab`, entonces se crean dos archivos. Cada lunes, miércoles, viernes y domingo, el primer trabajo del `cron` se ejecutará a las 2:30 a.m. para crear un backup de todo lo que hay en `/www`, y almacenará el archivo backup comprimido en el archivo `/backup/M-`

`W-F-Sun.tgz`. Del mismo modo, los martes, los jueves y los sábados por la mañana (a las 2:30 a.m.), la segunda entrada del `cron` creará un archivo con los mismos datos llamado `T-TH-Sat.tgz` en el mismo directorio `backup`. Tener dos backups le garantiza que tiene al menos dos backups de dos días en dos archivos comprimidos.

## Backup offline

También puede realizar backups en un medio transportable y guardarlas en localizaciones seguras. Este tipo de backup suele ser una operación que consume bastante tiempo. Para realizar este backup puede utilizar unidades de cintas, drives duros transportables (como los discos Jaz). Yo prefiero un backup basado en cintas de 8mm porque proporciona 8GB de capacidad de almacenaje de datos; además, llevan en el mercado mucho más tiempo que el nuevo medio transportable compacto.

A medida que sus sitios Web se enriquecen de contenido, el espacio Web disponible se llena rápidamente. Esto suele ser una consecuencia de archivos que están sin usar pero que nunca se han eliminado por miedo a que algo (como un enlace) se rompa en algún sitio. Si piensa que esto está sucediendo en su sitio Web, y está en una plataforma Unix, debería considerar ejecutar la utilidad `find` para localizar archivos a los que no se haya accedido durante mucho tiempo. Por ejemplo:

```
find /www -name "*.bak" -type f -atime +10 -exec ls -l {} \;
```

Esto da lugar a una lista con todos los directorios `/www` que terminan con la extensión `.bak` y a los que no se ha accedido desde hace 10 días. Si quiere eliminar estos archivos, puede reemplazar el comando `ls -l` y realizar una búsqueda como esta:

```
find /www -name "*.bak" -type f -atime +10 -exec rm -f {} \;
```

Si le sirve de ayuda, puede crear una entrada `cron` que ejecute este comando cada cierto tiempo.

## Definir estándares

Con un ciclo Web en marcha, tiene un entorno que puede acomodar muchos desarrolladores; sin embargo, únicamente creando un ciclo Web no está asegurando alta calidad de producción Web. Una alta calidad Web requiere contenido de alta calidad, y hay una serie de guías que puede seguir en cuanto al desarrollo de contenido. Se trata de definir sus estándares.

Cada sitio Web debería ofrecer contenido único para hacerlo atractivo a potenciales visitantes. Todos los tipos de contenido Web pueden clasificarse en

contenido estático y contenido dinámico. El contenido estático se crea normalmente con archivos HTML, y el contenido dinámico suele ser la salida de un CGI o de otras aplicaciones del lado del servidor o del lado del cliente.

La mayoría de los sitios Web utilizan una mezcla de ambos tipos de contenido para publicar su información; por lo tanto, son necesarios los estándares tanto para el desarrollo de contenido estático como para el desarrollo de contenido dinámico.

## Política de desarrollo de documentos HTML

Aunque pueda proporcionar contenido estático de muchos modos, como un plain-text o un archivo PDF, la mayor parte de los sitios Web utilizan documentos HTML como almacén principal de información. Para guiar a sus autores de HTML, debería crear una política de desarrollo de HTML. A continuación tenemos una serie de guías que puede adaptar para su organización.

### Utilice siempre etiquetas HTML estándar

Los desarrolladores de HTML deberían utilizar siempre el último estándar de HTML. La utilización de HTML dependiente de navegador hará que la página tenga un aspecto formidable en un tipo de navegador y horrible en otro.

Por ejemplo, el siguiente código muestra un esqueleto HTML que satisface el estándar HTML mínimo.

```
<HTML>
<HEAD><TITLE> Título del documento </TITLE> </HEAD>
<BODY>
    Cuerpo del documento
</BODY>
</HTML>
```

Cada uno de sus documentos debería contener al menos estas etiquetas HTML.

### Guarde imágenes in-line junto con los documentos

Las imágenes in-line de un documento deberían residir en un subdirectorio del directorio de documentos. Las referencias de la fuente de estas imágenes deben ser relativas, de modo que si se mueve el documento de una localización a otra junto con el directorio de imágenes, la imagen sigue rindiendo exactamente igual que antes.

**NOTA:** Hay una excepción a esta regla: si alguna de sus imágenes es reutilizable, deberá considerar colocarla en un directorio central de imágenes. Un ejemplo de este tipo de casos es una barra de navegación estándar implementada utilizando archivos de imagen. La barra de navegación se

**puede reutilizar en varios documentos, de modo que debería guardar estas imágenes en un directorio central en lugar de guardarlas con cada documento. Esto proporciona mejor control y ahorra espacio en el disco.**

El siguiente ejemplo le muestra cómo crear un documento HTML transportable que tiene varios archivos de gráficos unidos a él. Imagine que quiere publicar dos documentos HTML (mydoc1.html y mydoc2.html) que contienen tres imágenes (image1.gif, image2.gif y image3.gif). Puede crear primero un subdirectorio significativo bajo su directorio raíz de documentos o bajo cualquier otro directorio que sea apropiado. Vamos a suponer que crea este directorio bajo el directorio raíz de documentos del servidor (/www/mycompany/htdocs) y que lo llama mydir.

Ahora, tiene que crear un subdirectorio de imágenes llamado images bajo el directorio mydir y almacenar sus tres imágenes en este directorio. Edite sus documentos HTML de modo que todos los enlaces a las imágenes utilicen el atributo SRC del siguiente modo:

```
SRC="images/image1.gif"  
SRC="images/image2.gif"  
SRC="images/image3.gif"
```

Un ejemplo de un enlace a una imagen in-line para image3 sería:

```
<IMG SRC="images/images3.gif" HEIGHT="20" WIDTH="30" ALT="Image  
3 Description">
```

Los atributos SRC de la línea anterior no contienen ninguna información sobre rutas absolutas. Si se moviesen los documentos desde mydir a otherdir junto con el subdirectorio images, no habría ningún problema con las imágenes. Sin embargo, si los enlaces contienen información de rutas del estilo a:

```
<IMG SRC="mydir/images/images3.gif" HEIGHT="20" WIDTH="30"  
ALT="Image 3 Description">
```

o

```
<IMG SRC="/mydir/images/images3.gif" HEIGHT="20" WIDTH="30"  
ALT="Image 3 Description">
```

tendríamos que determinar estos documentos una vez movidos. Muchos sitios guardan sus imágenes en un directorio central de imágenes (como images) bajo la raíz de documentos y enlazan los documentos utilizando etiquetas IMG como:

```
<IMG SRC="/images/images3.gif" HEIGHT="20" WIDTH="30"  
ALT="Image 3 Description">
```

Esto resulta perfecto, pero cuando quiere eliminar el documento HTML, necesita asegurarse de que también elimina la imagen apropiada en el directorio cen-

tral de imágenes. Si falla en este proceso, a la larga desaparecerá una gran cantidad de espacio en el disco en su cantera de imágenes. Por lo tanto, no es una buena idea mantener imágenes en un directorio central. Debería guardar imágenes en un subdirectorio con sus enlaces.

## Desplegar mensajes copyright en cada documento

Cada documento debería contener un mensaje copyright embebido (comentando) que nombre, con claridad, al dueño del documento y a todas sus imágenes. Debería aparecer también un mensaje parecido de copyright en cada página. Para facilitar la actualización del mensaje de copyright, debería considerar la utilización de una directiva SSI como la siguiente:

```
<!--#include file=/copyright.html" -->
```

Ahora, todo lo que necesita hacer es crear una página HTML llamada `copyright.html`, y colocarla bajo su directorio raíz de documentos. Como el contenido de esta página HTML se inserta en el documento compatible con SSI que hace esta llamada, no necesita utilizar las etiquetas `<HTML>`, `<HEAD>`, `<TITLE>` o `<BODY>`. Utilizar las llamadas SSI le hará la vida más sencilla cuando necesite actualizar el año en el mensaje copyright, o cuando necesite realizar cualquier otro cambio.

## Política de desarrollo de aplicaciones dinámicas

El contenido dinámico se produce normalmente con scripts CGI o con otras aplicaciones que implementen CGI o alguna interfaz del lado del servidor. Una gran mayoría de contenido se produce utilizando scripts CGI basados en Perl. Como los scripts y las aplicaciones CGI tienen normalmente poca vida de expansión, muchos desarrolladores de CGI no son partidarios de producir aplicaciones de alta calidad.

Si tiene pensado utilizar los scripts y las aplicaciones basadas en `FastCGI` o en `mod_perl`, es importante que se desarrollen del modo adecuado. Debería considerar las siguientes políticas cuando implemente scripts y aplicaciones para su contenido dinámico.

### Utilice siempre un control de la versión

Los desarrolladores de CGI deben utilizar un control de versión, que les permita volver a una versión o a una aplicación antiguas en caso de que la nueva versión contenga un error.

En la mayoría de los sistemas Unix, puede utilizar el software Concurrent Versions System (CVS) para implementar un entorno de versiones controladas. Puede encontrar la última versión del software CVS en `ftp://prep.ai.mit.edu`.

## **No utilice nombres de rutas absolutos en los scripts ni en las aplicaciones CGI**

No se deben utilizar nombres de rutas en los scripts CGI. Esto asegura que los scripts se pueden utilizar en varios sitios Web sin modificación ninguna. Si se necesitan los nombres de ruta para un propósito determinado, se puede suministrar un archivo de configuración para el script; de este modo, las rutas se pueden actualizar modificando el archivo de configuración.

## **Proporcionar documentación de usuario y de código**

Se necesita código fuente para estar bien documentado de modo que los desarrolladores futuros puedan actualizar los scripts sin perder mucho tiempo tratando de imaginar cómo funciona.

## **Evitar las etiquetas HTML embebidas en scripts o en aplicaciones**

Las salidas de los scripts CGI deberían estar manejados por plantillas. En otras palabras, un script CGI lee una salida de una plantilla y reemplaza los campos de datos dinámicos (que se pueden representar utilizando etiquetas personalizadas). Esto hace sencilla la actualización de las salidas de páginas para los desarrolladores de HTML, ya que el HTML no está dentro del script CGI. De hecho, los scripts CGI deberían contener la menor cantidad de HTML como sea posible.

## **No confiar en los datos introducidos por el cliente**

Para reducir los riesgos de seguridad, compruebe los datos introducidos por el usuario antes de utilizarlos. Puede aprender más sobre la verificación de entradas en el capítulo 18, en el que se discuten los riesgos relacionados con las entradas y las soluciones en detalle.

## **Evitar las variables globales en los scripts CGI basados en Perl**

Cuando desarrollamos scripts CGI en Perl, debería evitar las variables globales. Limitar el alcance de una variable es un modo de eliminar comportamientos impredecibles en los scripts. Los programadores de Perl deberían utilizar el siguiente código para las declaraciones de variables:

```
my $variable;
```

en lugar de :

```
local $variable;
```

porque la primera declaración crea una variable que sólo está disponible en el ámbito en el que se ha creado. La última definición simplemente crea una instancia local de una variable global, lo que supone una gran confusión. Perl 6 proba-

blemente cambiará la palabra clave ‘local’ por ‘temp’ para aclarar este concepto a los programadores.

## Proporcionar a su sitio Web una interfaz intuitiva

Utilizar HTML estándar y scripts y aplicaciones CGI bien escritas puede garantizar que su sitio Web sea mejor que muchos otros sitios Web. Sin embargo, hay otro aspecto del diseño de un sitio Web que tiene que considerar, la interfaz de usuario.

Piense en un sitio Web como si fuera una aplicación interactiva con un Graphical User Interface (GUI) visible en un navegador Web. La GUI necesita ser intuitiva para que la gente tenga una experiencia agradable mientras que están visitando su sitio Web.

En esta sección se discuten los puntos claves en el desarrollo de una GUI intuitiva. Además de hacer intuitiva su GUI, tiene que estar atento a enlaces rotos o a solicitudes a archivos eliminados. Utilice los registros de error de su servidor para detectar este tipo de problemas. Debería facilitar a los visitantes la interacción con su sitio. La mayor parte de los sitios utilizan una interacción sencilla basada en un formulario HTML o con un script CGI. Puede desarrollar cualquiera de estas interacciones. Proporcionar interacción es un buen modo de aprender lo que piensan sus visitantes sobre su sitio Web.

## Facilite la navegación en su sitio

Los usuarios deben ser capaces de ir de una página a otra con facilidad. Deberían ser capaces de localizar botones o barras de menú que les permitan moverse atrás y delante, o saltar a información relacionada.

Muchos diseñadores de páginas Web opinan que los navegadores Web populares incluyen botones de atrás y delante, por lo que es una redundancia incluirlos en la página. Esto es incorrecto. Imagine que un usuario encuentra una de sus páginas (distinta de la página de inicio) desde los resultados de un buscador. El usuario simplemente estaba realizando una búsqueda con una o más palabras clave, y el buscador le proporcionó una URL a una página de su sitio. El usuario está muy interesado en conocer algo más sobre su sitio, de modo que quiere empezar desde el principio del documento, pero no hay modo de que el usuario pueda hacerlo, porque el botón de Atrás del navegador le devuelve al buscador y le saca de la página.

Si esta página tuviese un enlace (o un botón) a una página anterior, el usuario podría navegar por su sitio Web sin problemas. Los diseñadores de páginas Web a los que no les gusta los botones extra insisten en que el usuario podría simple-

mente manipular la URL para llegar a la página de inicio y empezar desde allí. Bien, para llevar esto a cabo hay que suponer que existe un enlace claro a esa página (la que coincide con la palabra buscada) desde la página de inicio, lo cual no es siempre cierto.

Es una buena idea implementar una barra de menú que permita al usuario ir hacia atrás y hacia delante, y que también permita al usuario saltar a una localización relacionada, o incluso a la página de inicio.

## **Crear un diseño atractivo**

Piense en los sitios Web como en presentaciones llenas de colorido e interactivas, que están activas las 24 horas del día. Si la presentación no es correcta, sus visitantes abandonarán su sitio. Considere las siguientes guías para el desarrollo de un diseño Web atractivo.

### **Colores apropiados**

Asegúrese de no excederse con la elección de colores. La utilización de colores extremos hace de su sitio Web un sitio poco profesional y poco lustroso. Ha de preocuparse por el colorido y utilizar el esquema apropiado de colores. Por ejemplo, si su sitio Web está relacionado con juguetes para niños, debería ser bastante colorido. Si su sitio está relacionado con precios de procesadores de señales digitales, no va a necesitar colores brillantes ni llamativos.

### **Tamaño apropiado de texto**

Intente que el contenido principal aparezca en una fuente normal. La utilización de una fuente especial mediante <FONT FACE="mi fuente especial"> puede hacer que la página tenga una buena presentación en su navegador (porque tiene la fuente), pero en otro navegador, la página puede ser totalmente distinta y podría ser difícil de leer. Además, ha de tener cuidado con el tamaño del texto; no puede ser ni muy pequeño ni muy grande. Recuerde que si sus visitantes no pueden leer lo que tiene que decir en su página Web, no serán capaces de entender lo que quiere decir.

### **Mínima utilización de imágenes y animaciones**

Tenga cuidado con las imágenes innecesarias. Las imágenes en sus páginas Web hacen que se carguen con más dificultad. Recuerde que no todo el mundo está conectado a su sitio Web mediante una ADSL o una RDSI; la mayor parte de la gente sigue utilizando modems de 56K o 28.8K para su conexión a Internet. Una descarga lenta de una página Web puede hacer que un cliente potencial abandone su sitio.

Además, es necesario ser prudente con la utilización de información. Incluso las animaciones más sugerentes se convierten en aburridas después de las primeras visitas, de modo que asegúrese de no super poblar sus páginas con ellas.

# Elimine los mensajes de error en clave

Configure Apache con la directiva ErrorDocument, de modo que los usuarios no reciban mensajes de error en el servidor que sean difíciles de entender (al menos para el usuario medio). Por ejemplo, cuando no se encuentra una solicitud URL en el servidor, el servidor desplegará un mensaje de error en clave. Para hacer este mensaje de error más intuitivo, puede añadir una directiva ErrorDocument como la siguiente:

```
ErrorDocument 404 /sorry.html
```

en el archivo httpd.conf, de modo que los visitantes de la Web puedan entender el mensaje de error.

## Pruebe su GUI Web

Uno de los mejores modos de probar su interfaz Web es utilizar un sistema que se parezca al ordenador medio de los usuarios de Internet, o quizás al ordenador de un cliente potencial. Si piensa que sus clientes van a tener ordenadores de alto rendimiento con conexiones rápidas, no tiene que preocuparse por la utilización de pocos gráficos o de aplicaciones del lado del cliente como son los applets de Java y las animaciones Shockwave.

En la mayoría de los casos, no conoce las especificaciones del ordenador ni de la red de sus potenciales clientes, de modo que debería centrarse en el sistema del usuario medio. Utilice un ordenador Pentium low-end con 16MB de RAM y una conexión vía modem de 28.8K para probar su sitio Web desde una cuenta ISP. Pruebe con bajas resoluciones para el monitor, como 640x480 o 800x600 píxeles; si sus visitantes objetivo utilizan sistemas Web-TV, pruebe con una resolución de 550x400 píxeles.

Si disfruta navegando por su sitio Web, otros probablemente también disfrutarán. Por otro lado, si no le gusta lo que ve, a otros tampoco les gustará.

**TRUCO:** Si lo prefiere, puede realizar una prueba de su sitio Web con productos de terceros. Por ejemplo, Netscape.com proporciona un servicio gratuito de ayuda basado en la Web, que puede encontrar en <http://webpage.netscape.com>. Esta aplicación de Netscape puede examinar cualquier sitio Web en relación con el tiempo de descarga de páginas, calidad del HTML, enlaces que terminan en puntos muertos, errores de ortografía, calidad del diseño HTML y la popularidad del enlace. Para ponerlo a prueba, simplemente diríjase al sitio Web anterior e introduzca su propia dirección del sitio Web y su dirección de correo electrónico. El tiempo de unos segundos a unos minutos. Recibirá un diagnóstico gratuito de su sitio Web.

## Promocionar su sitio Web

¿Qué tiene de bueno un sitio Web perfecto si nadie lo conoce? Debería pensar en promocionar su sitio Web en la Web. Puede alquilar agencias de publicidad para ayudarle en este asunto, aunque los anuncios en la Web pueden resultar caros. Si su presupuesto no es muy alto puede hacer algo para promocionarse a sí mismo. La siguiente lista le ofrece algunos puntos para promocionar su sitio adecuadamente.

- **Buscadores:** antes de hacer nada para promocionar su sitio Web, pregúntese, "¿cómo encuentro información en la Web?" La respuesta es: a través de los buscadores. ¿Está su compañía en una lista de un buscador? Si la respuesta es no, este es el primer paso en la promoción de su sitio Web. Casi todos los buscadores le permiten introducir su URL en la base de datos del robot de búsqueda, de modo que este robot puede atravesar su Web en el futuro. Debería realizar una lista de los buscadores que considere más importantes, e introducir la URL de su sitio Web en estos buscadores. Este proceso puede llevarle días o semanas.
- **Etiquetas META:** puede añadir información META en su contenido para que su URL aparezca en una posición decente cuando un cliente potencial realiza una búsqueda. Por ejemplo, puede añadir información META del tipo:

```
<META NAME="KEYWORD" CONTENT="palabraclave1 palabraclave2  
palabraclave3 ...">  
<META NAME="DESCRIPTION" CONTENT="Descripción de su  
compañía">
```

- **Intercambio de enlaces:** para incrementar el tráfico en su sitio web, puede participar en el intercambio de enlaces en sitios como [www.linkexchange.com](http://www.linkexchange.com). Los intercambios de enlaces requieren que ponga un conjunto especial de etiquetas HTML en sus páginas Web; estas etiquetas introducen archivos de gráficos de publicidad (banner) en sus páginas Web. Como contrapartida, sus gráficos de publicidad se mostrarán en otros sitios Web. Este tipo de compartición de publicidad es muy popular entre personas y en los negocios de pequeño tamaño.

Si compra un espacio de publicidad en sitios Web de alto perfil como Yahoo, AltaVista o Netscape, o utiliza el método de intercambio de enlaces, debería comprobar periódicamente que su sitio se encuentra en las salidas de los buscadores generando sus propias peticiones.



# **Parte III**

# **Ejecutar**

# **aplicaciones**

# **Web**



# 12 Ejecutar scripts CGI

---

## En este capítulo

1. Entendemos los conceptos básicos del Common Gateway Interface.
2. Configuramos Apache para CGI.
3. Proporcionamos acceso a `cgi-bin` a usuarios individuales.
4. Ejecutamos aplicaciones CGI utilizadas habitualmente.
5. Configuramos Apache para depurar las aplicaciones CGI.

El contenido dinámico dirige la Web. Sin contenido dinámico y personalizable, la Web resultaría un sitio del tipo "se encuentra en tal sitio, haga tal cosa". Después de todo, la gente no va a buscar y a experimentar el mismo contenido caduco una y otra vez. El contenido dinámico se está convirtiendo en una realidad gracias a la ayuda de una especificación llamada Common Gateway Interface (CGI). La especificación CGI le dice al servidor Web cómo interaccionar con aplicaciones externas. Un servidor Web que ejecuta aplicaciones CGI, le permite, prácticamente a cualquiera, ejecutar a demanda una lista seleccionada de programas en su servidor. Este capítulo trata los conceptos básicos del CGI para

ofrecerle una compresión clara sobre el tema, y los detalles la preparación de Apache para soportar ejecuciones CGI.

## ¿Qué es CGI?

Para proporcionar contenido dinámico e interactivo en la Web, muchos sitios Web populares utilizan aplicaciones CGI. Lo cierto es que usted ha utilizado ya algunas aplicaciones CGI en la Web. Por ejemplo, cuando rellena un formulario Web es muy probable que éste se procese mediante un script CGI escrito en Perl o en algún otro lenguaje.

Por supuesto, a medida que surgen más tecnologías Web, aparecen nuevas formas de distribuir contenido dinámico en la Web. La mayoría de estas soluciones son específicas para un lenguaje determinado, o dependen de un sistema operativo o del desarrollo de un software comercial. CGI, por su parte, es una especificación de una interfaz de puente (gateway), independiente del lenguaje que se puede implementar utilizando de forma cualquier lenguaje de desarrollo de aplicaciones conocido, incluyendo C, C++, Perl, lenguajes de script shell y Java.

Esta sección le ofrece un resumen del funcionamiento de un programa CGI (ver la figura 12.1). La idea principal es que el servidor Web obtiene una URL determinada, que de forma casi mágica, al menos por ahora, le dice al servidor que debe ejecutar una aplicación externa llamada `helloworld.cgi`. El servidor Web lanza la aplicación, espera a que se complete y devuelve un resultado. Entonces, transmite el resultado de la aplicación al cliente Web del otro lado.

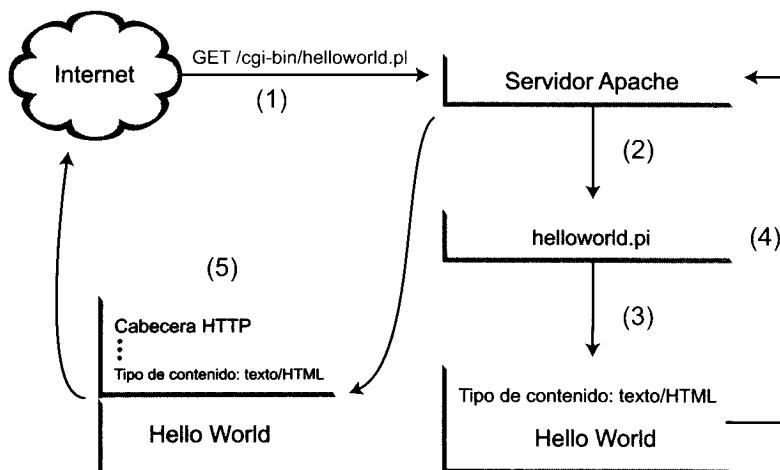


Figura 12.1. Funcionamiento de un programa CGI

En ocasiones vamos a necesitar que el cliente sea capaz de interaccionar con la aplicación. La aplicación debe suministrar los datos introducidos por el cliente. Por otro lado, cuando una aplicación produce una salida, ¿cómo saben el servidor

o el cliente el tipo de resultado que devuelve? Un programa puede producir un mensaje de texto, un formulario HTML para entradas, una imagen, y otro tipo de contenido. Como puede ver, el resultado puede variar mucho de una aplicación a otra, de modo que debe existir una forma para que las aplicaciones informen al servidor Web y al cliente sobre el tipo de resultado.

CGI define un conjunto de significados estándar para que éste le pase al cliente las salidas de las aplicaciones externas, y además define las formas en las que una aplicación externa puede devolver un resultado. Cualquier aplicación que se adhiera a estos estándares, se pueden etiquetar como una aplicación, programa o script CGI. Por simplicidad, utilizo el término programa CGI para referirme a cualquiera de ellos (como un script Perl o un programa C) que sea compatible con una especificación CGI. En la siguiente sección, veremos cómo trabaja un proceso de entrada/salida CGI.

## Input y Output CGI

Hay muchas formas en las que un servidor Web puede recibir información de un cliente (un navegador Web, por ejemplo). El protocolo HTTP define el modo en el que un servidor Web y un cliente pueden intercambiar información. Los métodos más comunes para trasmitir solicitudes a un servidor Web son las solicitudes GET y las solicitudes POST, que se van a describir en las secciones siguientes.

### Solicitudes GET

Las solicitudes GET son el método más sencillo de enviar solicitudes HTTP. Cada vez que introduce una dirección de un sitio Web en su servidor, genera una solicitud GET y la envía al servidor Web solicitado. Por ejemplo, si introduce `http://www.hungryminds.com` en su navegador Web, envía una solicitud HTTP como la siguiente:

```
GET /
```

al servidor Web `www.hungryminds.com`. Esta solicitud GET le pide al servidor Web de Hungry Minds que devuelva el documento de máximo nivel del árbol de documentos de la Web. Este documento se llama normalmente página de inicio, y normalmente se refiere a la página `index.html` del directorio Web de máximo nivel. Además, HTTP le permite codificar información adicional en una solicitud GET. Por ejemplo:

```
http://www.mycompany.com/cgi-bin/search.cgi?books=cgi&author=kabir
```

Aquí, la solicitud GET es:

```
GET www.mycompany.com/cgi-bin/search.cgi?books=cgi&author=kabir
```

Esto le dice al servidor que ejecute el programa CGI /cgi-bin/search.cgi y que le pase los datos introducidos, book=cgi y author=kabir.

Cuando un servidor compatible con CGI como Apache, recibe este tipo de solicitud, sigue las especificaciones CGI y pasa los datos introducidos a la aplicación (en este caso, search.cgi en el directorio cgi-bin). Cuando un recurso CGI se solicita mediante el método GET de solicitudes HTTP, Apache:

1. Asigna las variables de entorno al programa CGI, que incluye almacenaje del nombre del método de solicitudes HTTP en una variable de entorno llamada REQUEST\_METHOD, y el dato recibido del cliente en una variable de entorno llamada QUERY\_STRING.
2. Ejecuta el programa CGI solicitado.
3. Espera a que se complete el programa y devuelva una salida.
4. Analiza la salida del programa CGI si se trata de un programa de cabecera no analizada (un programa CGI con la cabecera sin analizar crea sus propias cabeceras HTTP de modo que el servidor no necesita analizar las cabeceras).
5. Crea la cabecera o las cabeceras HTTP necesarias.
6. Envía las cabeceras y el resultado del programa al cliente que las ha solicitado. La figura 12.2 ilustra este proceso.

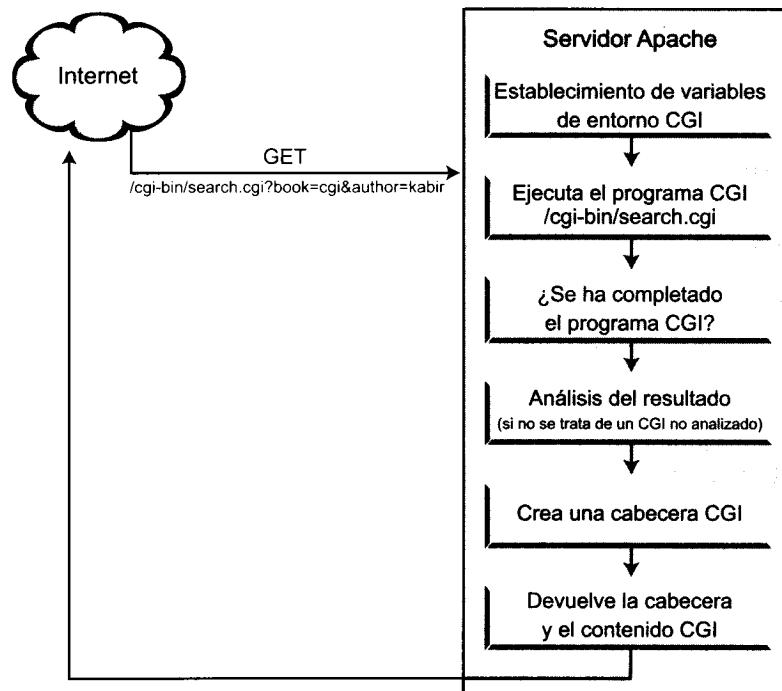


Figura 12.2. Proceso CGI en un servidor

A continuación vamos a ver lo que tiene que hacer un programa CGI para recuperar los datos introducidos para utilizarlos para propósitos internos.

Tal y como muestra la figura 12.3, un programa CGI.

1. Lee la variable de entorno REQUEST\_METHOD.
2. Determina si se utiliza o no el método GET utilizando la variable almacenada en la variable REQUEST\_METHOD.
3. Si se utiliza el método GET, recupera el dato almacenado en la variable de entorno QUERY\_STRING, .
4. Codifica el dato.
5. Procesa el dato codificado.
6. Escribe el tipo de contenido del resultado en su herramienta de salida estándar (STDOUT), tras completar el proceso.
7. Escribe el dato resultante en el STDOUT y sale.

El servidor Web lee el STDOUT de la aplicación y lo analiza para localizar el tipo de contenido de la salida. Entonces transmite las cabeceras HTTP y el Content-Type apropiados antes de transmitir la salida al cliente. El programa CGI se abandona y se completa la transacción CGI.

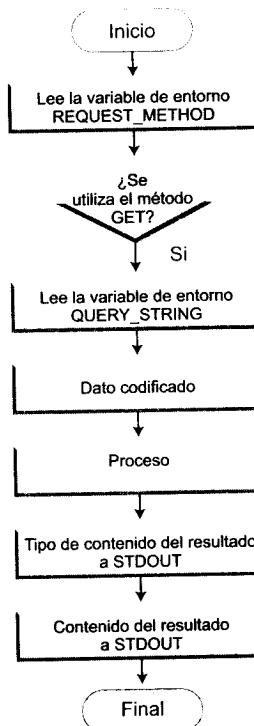


Figura 12.3. Proceso CGI en el servidor

**NOTA:** Si tenemos un programa CGI para proporcionar toda la información sobre las cabeceras HTTP y el tipo de contenido, hay que ponerle el prefijo nph (para cabeceras las cabeceras no analizadas) a su nombre. El servidor no realiza la salida nph de un programa CGI y la transmite directamente al cliente; la mayoría de los programas CGI dejan que el servidor escriba la cabecera HTTP y, por lo tanto, se trata de programas de cabeceras analizadas.

La utilización del método GET para pasar datos introducidos en un programa CGI, está limitado en muchos sentidos, incluidos los que se exponen a continuación:

- El tamaño total de datos que se pueden transmitir como parte de una URL está limitado por el límite de la longitud de la URL del cliente. Muchos navegadores Web tienen límites de hardware para la longitud de una URL, y por lo tanto, el número total de datos que se puede enviar mediante una URL codificada es muy pequeño. Sin embargo, en ocasiones puede ser una gran idea pasar datos a los programas CGI mediante una URL. Por ejemplo, si tiene un formulario HTML que utiliza el método GET para enviar datos a un programa CGI, esta URL se puede introducir en la carpeta de favoritos para utilizarla más tarde, sin necesidad de volver a introducir los datos de entrada. Esto puede resultar muy cómodo para las aplicaciones de consulta a bases de datos.
- La longitud del valor de una sola variable de entorno (QUERY\_STRING) está limitada. Muchos, sino todos, sistemas operativos tienen límites en el número de bytes que puede contener el valor de una variable de entorno. Esto limita el número total de bytes que se pueden almacenar como datos de entrada.

Estos límites no influyen en los programas CGI que necesitan pocas entradas de usuario o que no las necesitan. Para aquellos programas que necesitan una gran cantidad de datos introducidos por el usuario, sin embargo, es mejor utilizar otro método (el método POST) para las solicitudes HTTP. El método POST se discute en la siguiente sección.

## Solicitudes POST

El método POST se utiliza para pasar datos a programas CGI. Suele encontrar este método en los formularios HTML que rellena en las páginas Web. Como ejemplo, tenemos el listado 12.1.

Observe que hay una sección <FORM> </FORM> en el listado. Un formulario HTML tiene normalmente una etiqueta de inicio <FORM> que define la acción ACTION y el método METHOD para el formulario. En el ejemplo anterior,

la acción es el programa CGI /cgi-bin/search.cgi y el método es el método POST.

#### Listado 12.1. Un formulario HTML utilizando el método POST

```
<HTML>
  <HEAD>
    <TITLE> Apache Server 2.0 - Listado capitulo 12 12.1 </
    TITLE>
  </HEAD>

  <BODY>
    <H1>Listing 12-1</H1>
    <H2>Un ejemplo de un formulario HTML utilizando el método POST </
    H2>
    <HR>
    <FORM ACTION="/cgi-bin/search.cgi" METHOD="POST">

    <PRE>
      Tipo de libro <INPUT TYPE="TEXT" NAME="book" SIZE="10"
      MAXSIZE="20">
      Nombre del autor <INPUT TYPE="TEXT" NAME="author" SIZE="10"
      MAXSIZE="20">
    </PRE>
    <INPUT TYPE=SUBMIT VALUE="Search Now">

  </FORM>
</BODY>
</HTML>
```

Después de la etiqueta <FORM>, normalmente hay una o más entidades INPUT; las entidades INPUT deben incluir cajas de texto, menús desplegables y listados. En nuestro ejemplo, hay tres entidades de introducción de datos. La primera permite al usuario introducir un valor para la variable book. La siguiente es parecida y permite al usuario introducir un valor para la variable author, y la última es algo distinta y permite al usuario enviar el formulario. Cuando el usuario envía el formulario, el software del cliente transmite una solicitud POST al servidor para el recurso ACTION (es decir, /c/s.dll/search.cgi), y además transmite los valores book=<valor introducido por el usuario> y author=<valor introducido por el usuario> en un formato codificado.

## Comparar GET y POST

¿Cuál es la diferencia entre las solicitudes GET y POST? Los datos enviados con el método POST no se almacenan en la variable de entorno QUERY\_STRING de un programa CGI. Por el contrario, se almacenan en la entrada estándar

(STDIN) del programa CGI. La variable REQUEST\_METHOD se asigna a POST, mientras que los datos codificados se almacenan en el STDIN del programa CGI, y se asigna una nueva variable de entorno llamada CONTENT\_LENGTH al número de bytes almacenados en el STDIN.

El programa CGI debe comprobar el valor de la variable de entorno REQUEST\_METHOD. Si está fijado en POST para las solicitudes HTTP POST, el programa debería determinar primero el tamaño de los datos introducidos con el valor de la variable de entorno CONTENT\_LENGTH y, entonces, leer los datos desde el STDIN. Observe que el servidor Web no es responsable de la inserción de un marcador End-of-File (EOF) en el STDIN, que es por lo que la variable CONTENT\_LENGTH está asignada a la longitud de los datos, en bytes, haciendo más fácil que el programa CGI determine el número total de bytes en los datos.

Es posible utilizar GET y POST al mismo tiempo. A continuación tenemos un ejemplo de un formulario HTML que utiliza oficialmente el método POST, pero que además utiliza una cadena de consulta, `username=joe`, como parte del ACTION CGI.

```
<FORM ACTION="/cgi-bin/edit.cgi?username=joe" METHOD=POST>
<INPUT TYPE=TEXT NAME="PhoneNumber">
</FORM>
```

En este ejemplo, la consulta `username=joe` formará parte de la URL, pero el otro campo (`PhoneNumber`) formará parte del dato POST. El efecto: el usuario final puede introducir la URL en la carpeta de favoritos y ejecutar el script `edit.cgi` como `joe` sin asignar valores para ningún otro campo. Esto resulta perfecto para las aplicaciones de bases de datos online y para los buscadores.

Tanto si utiliza GET, o POST, o ambas, los datos son codificados y devueltos al programa CGI para decodificarlos. La siguiente sección discute los conceptos implicados en la decodificación de los datos.

## Decodificación de los datos introducidos

Los diseñadores del protocolo HTTP planearon una implementación sencilla del protocolo en cualquier sistema. Además, realizaron un esquema de decodificación de los datos muy sencillo. Los esquemas definen ciertos caracteres como caracteres especiales. Por ejemplo, el signo igual (=) facilita la realización de pares clave=valor; el signo más (+) reemplaza el carácter espacio, y el carácter ampersand (&) separa dos pares clave=valor.

Si el dato contiene caracteres con un significado, debería preguntarse qué es lo que se está transmitiendo. En este caso, se utiliza un esquema de codificación de tres caracteres. Un signo de porcentaje (%) indica el comienzo de una secuencia codificada de caracteres que consiste en dos dígitos hexadecimales.

El sistema hexadecimal es un sistema de numeración en base 16 en el que los números del 0 al 9 representan los mismos valores que los números del 0 al 9 en

el sistema decimal, pero tiene un conjunto extra de dígitos. Estos dígitos extra son A (=10), B (=11), C (=12), D (=14) y F (=15). Por ejemplo, el 20 en el sistema hexadecimal es igual al 32 en el sistema decimal. El sistema de conversión es:

$$20 = 2 \times (16^1) + 0 \times (16^0)$$

Estos dos dígitos consisten en el valor que puede integrar en la tabla ASCII (para el inglés) para obtener el carácter. Por ejemplo, %20 (hexadecimal) es 32 (decimal) y se corresponde al carácter espacio en la tabla ASCII.

## Variables CGI Apache

Hay dos modos en los que Apache puede implementar soporte CGI. La distribución estándar de Apache incluye un módulo CGI que implementa el soporte CGI tradicional; sin embargo, hay un nuevo módulo (*FastCGI*) que implementa soporte para aplicaciones CGI de alto rendimiento. Esta sección discute el soporte estándar CGI.

En las secciones anteriores, vimos que un servidor Web que es compatible con CGI, utiliza variables de entorno, entradas estándar (*STDIN*) y salidas estándar (*STDOUT*) para transferir información a y desde los programas CGI. Apache proporciona un conjunto flexible de variables de entorno para los desarrolladores de programas CGI.

Utilizando estas variables de entorno, un programa CGI no sólo recupera datos de entrada, sino que, además, reconoce el tipo de cliente y servidor con el que está tratando.

En las siguientes secciones, veremos las variables de entorno que están disponibles desde el módulo CGI estándar compilado en Apache.

**NOTA:** La distribución del código fuente de la versión 2.x.x de Apache, soporta la opción `--enable-cgid` para el script `configure`. Esta opción fuerza a Apache a utilizar un servidor de scripts (llamado demonio CGI) para manejar los procesos del script CGI, los cuales aumentan el rendimiento general de Apache.

## Variables del servidor

Estas variables son asignadas por Apache para informar a los programas CGI sobre Apache. Utilizando variables del servidor, un programa CGI puede determinar información específica sobre varios servidores, como la versión del software de Apache, la dirección de correo electrónico del administrador, e información de este tipo.

## **SERVER\_SOFTWARE**

SERVER\_SOFTWARE está asignado por Apache, y el valor se encuentra habitualmente de la siguiente forma:

Apache/Version (OS Info)

En este caso, Apache es el nombre del software del servidor que está ejecutando el programa CGI, y version es el número de la versión de Apache. Un valor de ejemplo sería:

Apache/2.0.14 (Unix)

Esto resulta útil cuando un programa CGI se utiliza para sacar partido de una característica nueva que se encuentra en la última versión de Apache, y sigue siendo capaz de rendir en versiones antiguas.

GATEWAY\_INTERFACE le dice al programa CGI qué versión de la especificación CGI soporta actualmente el servidor. Un valor de ejemplo sería:

CGI/1.1

Un programa CGI puede determinar el valor de esta variable y, de forma condicional, hacer uso de las distintas características disponibles en las diferentes versiones de las especificaciones CGI. Por ejemplo, si el valor es CGI/1.0, el programa no utilizará ninguna característica CGI/1.1, o viceversa.

El primer entero antes de la coma decimal se llama número principal, y el entero que se encuentra detrás de la coma es el número secundario. Como estos dos enteros se tratan como números separados, CGI/2.2 es una versión más antigua que CGI/2.15.

## **SERVER\_ADMIN**

Si utiliza la directiva ServerAdmin en el archivo httpd.conf para determinar la dirección de correo electrónico del administrador del sistema, esta variable será asignada para reflejarlo. Además, observe que si tiene una directiva ServerAdmin en un contenedor de configuración de un host virtual, la variable SERVER\_ADMIN se asigna a esa dirección en el caso de que el programa CGI al que se está accediendo sea parte del host virtual.

## **DOCUMENT\_ROOT**

Esta variable está asignada al valor de la directiva DocumentRoot del sitio Web al que se está accediendo.

## **Variables para las solicitudes del cliente**

Apache crea un conjunto de variables de entorno desde la cabecera de la solicitud HTTP que recibe desde una solicitud de un programa CGI. Proporciona

esta información al programa CGI creando el siguiente conjunto de variables de entorno.

## **SERVER\_NAME**

Esta variable le dice al programa CGI a qué host se está accediendo. El valor es una dirección IP o el nombre completo del host:

```
SERVER_NAME = 192.168.1.100  
SERVER_NAME = www.domain.com
```

## **HTTP\_HOST**

Ver la variable SERVER\_NAME.

## **HTTP\_ACCEPT**

Esta variable está asignada a la lista de tipos MIME que el cliente puede aceptar, incluidos los siguientes:

```
HTTP_ACCEPT = image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,  
image/png, */*
```

Aquí, el cliente dice que es capaz de manejar imágenes GIF, JPEG, PNG, y otro tipo de imágenes. Esto le permite al programa CGI determinar el tipo de salida ideal para el cliente.

Por ejemplo, un programa CGI podría producir GIF o JPEG y recibir una HTTP\_ACCEPT:

```
HTTP_ACCEPT = image/gif, */*
```

Entonces, puede enviar un resultado GIF en lugar de uno JPEG porque el cliente lo prefiere de ese modo.

## **HTTP\_ACCEPT\_CHARSET**

Esta variable determina el conjunto de caracteres aceptable para el cliente, por ejemplo:

```
HTTP_ACCEPT_CHARSET = iso-8859-1, *, utf-8
```

## **HTTP\_ACCEPT\_ENCODING**

Esta variable determina los esquemas de codificación aceptables para el cliente, por ejemplo:

```
HTTP_ACCEPT_ENCODING = gzip
```

En este caso, el cliente acepta archivos gzip (comprimidos). De ese modo, un script CGI puede comprimir una página grande utilizando la compresión gzip y enviándola después, y el cliente Web será capaz de descomprimirlo.

## **HTTP\_ACCEPT\_LANGUAGE**

Esta variable determina el lenguaje aceptable para el cliente, por ejemplo:

HTTP\_ACCEPT\_LANGUAGE = en

En este caso, el cliente acepta el contenido en inglés.

## **HTTP\_USER\_AGENT**

Esta variable determina qué software del cliente y qué sistema operativo está ejecutando el sistema, por ejemplo:

HTTP\_USER\_AGENT = Mozilla/4.04 [en] (WinNT; I)

Este código es equivalente al siguiente:

```
Client Software = Netscape Navigator ()  
Client Software Version = 4.04 (English version)  
Operating System = Windows NT (Intel)
```

Observe que Mozilla es una palabra clave utilizada por Netscape para la base de código de Navigator. Aunque únicamente los navegadores Netscape utilizan la palabra Mozilla, muchos otros fabricantes han comenzado a utilizar Mozilla como parte de la cabecera HTTP. Por ejemplo, Microsoft Internet Explorer (IE) 5.5 produce el siguiente dato HTTP\_USER\_AGENT cuando se ejecuta en la misma máquina:

HTTP\_USER\_AGENT = Mozilla/5.5 (compatible; MSIE 5.5; Windows NT)

Esta información sobre el agente usuario, se utiliza en muchos sitios Web. Un sitio que está optimizado para Netscape Navigator (es decir, que utiliza una característica de HTML, o JavaScript, o un plug-in, que funciona adecuadamente en Netscape Navigator) debería utilizar la información HTTP\_USER\_AGENT para devolver una página distinta para los usuarios que navegan en el sitio con IE, o para cualquier otro navegador menos conocido. Sin embargo, recomiendo que se adhiera al estándar HTML (la especificación HTML para el estándar actual, se encuentra disponible en [www.w3.org](http://www.w3.org)), y que no implemente ninguna característica específica de navegador.

Aunque optimizar sus páginas para un navegador concreto dará lugar a unas páginas perfectas en ese navegador, recuerde que no todo el mundo utiliza el mismo navegador. Esto significa que sus etiquetas HTML o sus plug-ins específicos dificultarán la visita a su sitio Web a todos aquellos que no utilicen su navegador Web preferido.

## **HTTP\_REFERER**

Esta variable está asignada al Uniform Resource Identifier (URI) que dirige las solicitudes al programa CGI al que se ha llamado. Utilizando esta variable, puede decidir si la solicitud proviene del enlace de una de sus páginas Web o de

una URI remota. Observe que un error de ortografía en el nombre de la variable da lugar a confusión en los desarrolladores de CGI, que lo están deletreando correctamente en sus aplicaciones y scripts, al descubrir que no están funcionando. Por eso, si ha pensado utilizar esta variable, deletréelo tal y como se indica aquí.

## **HTTP\_CONNECTION**

La variable `HTTP_CONNECTION` está asignada al tipo de conexión que utiliza el cliente y el servidor. Por ejemplo:

```
HTTP_CONNECTION = Keep-Alive
```

Esto indica que el cliente es capaz de manejar conexiones utilizando `Keep-Alive` y que las está utilizando.

## **SERVER\_PORT**

El valor de la variable `SERVER_PORT` le dice al programa CGI qué puerto del servidor se está utilizando para acceder al programa. Un ejemplo sería:

```
SERVER_PORT = 80
```

Si un programa CGI crea unas URL que se dirigen al servidor, sería útil incluir la dirección del puerto, que se encuentra como el valor de esta variable, en la URL.

## **REMOTE\_HOST**

La variable `REMOTE_HOST` informa a un programa CGI sobre la dirección IP o el nombre IP del cliente, del siguiente modo:

```
REMOTE_HOST = dsl-666.isp24by7.net
```

Observe que si el servidor Apache está compilado con la opción `MINIMAL_DNS`, no se asigna esta variable.

## **REMOTE\_PORT**

El cliente utiliza este número de puerto en el lado del cliente de la conexión socket:

```
REMOTE_PORT = 1163
```

No he encontrado aún la utilidad a esta variable.

## **REMOTE\_ADDR**

La variable `REMOTE_ADDR` es la dirección IP del sistema del cliente:

```
REMOTE_ADDR = 192.168.1.100
```

Observe que si el cliente está tras un firewall o un servidor proxy, la dirección IP almacenada en esta variable no tiene por qué ser la dirección IP del sistema del cliente.

## **REMOTE\_USER**

La variable **REMOTE\_USER** será asignada únicamente cuando el acceso al programa CGI requiera autenticación HTTP basic. El nombre de usuario utilizado en esta autenticación, se almacena en esta variable para el programa CGI. El programa CGI, sin embargo, no tendrá modo de identificar la contraseña utilizada para acceder a él. Si esta variable está asignada al nombre de usuario, el programa CGI puede suponer con seguridad que el usuario suministró la contraseña adecuada para el acceso.

## **SERVER\_PROTOCOL**

**SERVER\_PROTOCOL** es el protocolo y el número de versión que utiliza el cliente para enviar la solicitud para el programa CGI:

```
SERVER_PROTOCOL = HTTP/1.1
```

## **REQUEST\_METHOD**

La variable **REQUEST\_METHOD** asigna el método de solicitudes HTTP utilizado por el cliente para solicitar un programa CGI. Los valores habituales son: GET, POST y HEAD.

```
REQUEST_METHOD=GET
```

La entrada se almacena en la variable **QUERY\_STRING** cuando el método es GET. Cuando el método es POST, la entrada se almacena en el **STDIN** del programa CGI.

## **REQUEST\_URI**

La variable **REQUEST\_URI** determina el URI de la solicitud.

```
REQUEST_URI = /cgi-bin/printenv2
```

## **REMOTE\_IDENT**

**REMOTE\_IDENT** será asignada sólo si se asigna la directiva **IdentifyCheck**. Esta variable almacena la información de identificación devuelta por el **identd** (demonio de identificación) remoto. Como muchos sistemas no ejecutan este tipo de procesos demonio, **REMOTE\_IDENT** no debería considerarse seguro en la identificación de usuarios. Recomiendo utilizar esta variable en un entorno de intranet o de extranet en el que usted o su organización esté ejecutando un servidor **identd**.

## **AUTH\_TYPE**

Si un programa CGI se almacena en una sección del sitio Web en la que se requiere la autentificación para acceder, esta variable se asigna para especificar el método de autentificación utilizado.

## **CONTENT\_TYPE**

Esta variable especifica el tipo MIME de cualquier dato adjunto a la cabecera de la solicitud. Por ejemplo:

```
CONTENT_TYPE = application/x-www-form-urlencoded
```

Cuando utilizamos un formulario HTML y el método POST, puede especificar el tipo de contenido en el formulario HTML utilizando el atributo TYPE de la etiqueta <FORM>, del siguiente modo:

```
<FORM ACTION="/cgi-bin/search.cgi"
      METHOD="POST"
      TYPE= "application/x-www-form-urlencoded">
```

## **CONTENT\_LENGTH**

Cuando se utiliza el método POST, Apache almacena los datos introducidos (adjuntos a la solicitud) en el STDIN del programa CGI. El servidor no inserta un marcador End-of-File (EOF) en el STDIN. Sin embargo, a esta variable se le asigna el número de bytes. Por ejemplo, si:

```
CONTENT_LENGTH = 21
```

entonces el programa CGI en cuestión leería 21 bytes de datos desde su STDIN.

## **SCRIPT\_NAME**

SCRIPT\_NAME es el URI del programa CGI:

```
SCRIPT_NAME = /cgi-bin/search.cgi
```

## **SCRIPT\_FILENAME**

SCRIPT\_FILENAME es el nombre completo de la ruta del programa CGI:

```
SCRIPT_FILENAME = /www/kabir/public/cgi-bin/search.cgi
```

## **QUERY\_STRING**

Si un cliente Web, un navegador Web por ejemplo, utiliza el método GET y proporciona los datos introducidos tras una interrogación (?), el dato se almacena como el valor de esta variable. Por ejemplo, una solicitud para el siguiente programa CGI:

```
http://apache.domain.com/cgi-bin/search.cgi?key1=value1&key2=
value2
```

hará que Apache asigne:

```
QUERY_STRING = key1=value1&key2=value2
```

Línea que el programa CGI /cgi-bin/search.cgi puede leer y decodificar antes de utilizarla.

## **PATH\_INFO**

Si un dato introducido para un programa CGI forma parte del URI, la ruta extra (que es en realidad algún dato del programa al que se ha llamado) se almacena como el valor de la variable. Por ejemplo:

```
http://apache.domain.com/cgi-bin/search.cgi/argument1/argument2
```

hará que Apache asigne:

```
PATH_INFO = /argument1/argument2
```

**NOTA: PATH\_INFO no tiene nada que forme parte de la cadena de consulta. En otras palabras, si el URI incluye una cadena de consulta tras una ?, esta parte del dato se almacenará en la variable QUERY\_STRING. Por ejemplo:**

```
http://apache.domain.com/cgi-bin/search.cgi/CA/95825?book=apache&author=kabir
```

hará que Apache asigne las siguientes variables:

```
PATH_INFO = /CA/95825
```

```
QUERY_STRING=book=apache&author=kabir
```

## **PATH\_TRANSLATED**

Esta es la ruta absoluta del archivo solicitado. Por ejemplo, cuando un cliente Web, como un navegador Web, solicita `http://server/cgi-bin/script.cgi`, la ruta actual del script puede ser `path_to_cgi_alias/script.cgi`.

# **Configurar Apache para CGI**

Esta sección discute el modo de configurar Apache para procesar solicitudes CGI. El proceso de configuración incluye el decirle a Apache donde almacenar sus programas CGI, establecer manejadores CGI para extensiones de archivos determinadas, e indicar qué extensiones de archivos se consideran programas CGI. Es una buena idea mantener sus programas CGI en un directorio central. Esto le permite un control más adecuado de sus programas CGI. Mantener dis-

persos sus programas CGI por el espacio Web podría hacer que su sitio Web fuese imposible de manejar, y además puede crear agujeros de seguridad difíciles de detectar.

## Análisis del directorio de programas CGI

Establecer un directorio central para los programa CGI es el primer paso en el establecimiento de un entorno CGI seguro. Es mejor mantener este directorio central fuera de su directorio DocumentRoot de modo que un cliente Web, como por ejemplo un navegador, no pueda acceder directamente a los programas CGI. ¿Porqué? Porque de este modo no proporcionamos prácticamente información hacia fuera de nuestro sitio. Estamos proporcionando un nivel superior de seguridad a nuestro sitio o sitios. Hasta el hacker más novato sabe dónde se encuentran físicamente localizados sus programas CGI, y esto nos podría perjudicar.

Lo primero que necesita hacer es crear un directorio fuera de su directorio DocumentRoot. Por ejemplo, si /www/mycompany/public/htdocs es el directorio DocumentRoot de un sitio Web, entonces el alias para su directorio del programa CGI será, por ejemplo, /www/mycompany/public/cgi-bin. Para crear el alias para su directorio CGI, puede utilizar la directiva ScriptAlias. Si establece soporte CGI para el servidor Web principal, edite el archivo httpd.conf e inserte una línea ScriptAlias con la sintaxis:

```
ScriptAlias /alias/ "/fully/qualified/path/to/cgi/scripts/dir/"
```

Por ejemplo:

```
ScriptAlias /cgi-bin/ "/www/mycompany/public/cgi-bin/"
```

Si está asignando soporte CGI para un sitio virtual, añada una línea ScriptAlias en el contenedor <VirtualHost . . . > que defina el host virtual. Por ejemplo:

```
NameVirtualHost 192.168.1.100  
  
<VirtualHost 192.168.1.100>  
  
    ServerName blackhole.domain.com  
    DocumentRoot "/www/blackhole/public/htdocs"  
    ScriptAlias /apps/ "/www/blackhole/public/cgi-bin/"  
  
</VirtualHost>
```

Aquí el alias /apps/ se utiliza para crear un alias para el directorio del programa CGI. Si hay un programa CGI llamado feedback.cgi en el directorio /www/blackhole/public/cgi-bin, sólo se puede acceder a él mediante la siguiente URL:

<http://blackhole.domain.com/apps/feedback.cgi>

Una vez que asigna la directiva `ScriptAlias`, asegúrese de que el permiso del directorio permite a Apache leer y ejecutar los archivos que encuentre en el directorio.

El directorio asignado por `ScriptAlias` debería tener asignaciones de permisos estrictas. Únicamente el desarrollador del programa CGI o el administrador del sistema deberían tener permiso total (lectura, escritura y ejecución) para el directorio. Puede definir varios alias para los directorios de los programas CGI de modo que el directorio `ScriptAlias` especificado no sea navegable (por defecto) por razones de seguridad.

Cuando se realiza una solicitud a Apache, éste intenta ejecutar cualquier archivo ejecutable (con permiso de archivo) del directorio `ScriptAliased`. Por ejemplo:

```
http://blackhole.domain.com/apps/foo.cgi  
http://blackhole.domain.com/apps/foo.pl  
http://blackhole.domain.com/apps/foo.bak  
http://blackhole.domain.com/apps/foo.dat
```

Todas estas solicitudes URL avisarán a Apache para que intente ejecutar los distintos archivos `foo`.

## Elegir extensiones específicas de archivos CGI

No estoy absolutamente de acuerdo con la idea de que cualquier archivo en el directorio específico de `ScriptAlias` puede ejecutarse como un programa CGI. Prefiero la solución que me permite restringir los nombres del programa CGI de modo que únicamente los archivos con determinadas extensiones son tratados como programas CGI. La siguiente sección discute el modo en el que puede implementar esta solución utilizando un manejador Apache que puede encontrar en el módulo `mod_cgi` y que contiene una configuración de ejemplo, en la que he permitido que un conjunto seleccionado de extensiones de archivos sea tratado como programas CGI utilizando el manejador `AddHandler`.

Para este ejemplo, he supuesto que el nombre del servidor Apache es `www.domain.com`, y que su directorio `DocumentRoot` se encuentra en `/www/mysite/public/htdocs`; el directorio del programa CGI es `/www/nitec/public/cgi-bin`. Observe que el directorio del programa CGI está fuera del directorio específico de `DocumentRoot` de forma intencionada. Esto asegura que nadie pueda navegar por este directorio, y que sólo Apache pueda verlo mediante el alias.

Siga los pasos siguientes, para que Apache ejecute scripts CGI con una extensión o con extensiones determinadas para ejecutar desde un directorio:

1. Desactive cualquier directiva `ScriptAlias`, eliminándola totalmente del `httpd.conf`, o convirtiéndola en un comentario insertando una almohadilla (#) como primer carácter de la línea.

2. Tiene que crear un alias para el directorio del programa CGI. No hay modo de acceder al directorio del programa CGI sin un alias (o un enlace simbólico), ya que reside fuera del árbol de documentos. Puede definir un alias utilizando la directiva Alias, que tiene la sintaxis siguiente:

```
Alias /alias/ "/path/to/cgi/dir/outside/doc/root/"
```

Siguiendo esta sintaxis, la directiva Alias necesaria sería de este tipo:

```
Alias /cgi-bin/ "/www/mysite/public/cgi-bin/"
```

3. Instruya a Apache para que ejecute programas CGI desde este directorio, definiendo un contenedor <Directory> para este directorio especial. La definición del contenedor del directorio que es necesaria para hacer que todo esto ocurra (es decir, para convertir el directorio en un directorio para el programa CGI) es:

```
<Directory "/path/to/cgi/dir/outside/doc/root">
    Options ExecCGI -Indexes
    AddHandler cgi-script extension-list
</Directory>
```

La directiva Options asigna dos opciones para el directorio /path/to/cgi/dir/outside/doc/root. Primero se asigna la opción ExecCGI, que le dice a Apache que permita la ejecución de programas CGI desde este directorio. Después, la opción -Indexes le dice a Apache que desactive el listado de directorios ya que no es una buena idea permitir a los visitantes que vean el contenido del directorio del programa CGI. A continuación, la directiva AddHandler asigna el manejador de scripts CGI a una lista de extensiones de archivos encontrados en este directorio. Cualquier archivo con dichas extensiones es tratado como un programa CGI. Cuando un cliente Web realiza una solicitud para un archivo de este tipo, el programa se ejecuta y el resultado se devuelve al cliente Web. Las directivas reales para nuestro ejemplo actual son del siguiente estilo:

```
<Directory "/www/mysite/public/cgi-bin">
    Options ExecCGI -Indexes
    AddHandler cgi-script .cgi .pl
</Directory>
```

En este ejemplo, está permitiendo .cgi y .pl como extensiones de programas CGI y, por lo tanto, cuando se realizan solicitudes como estas:

```
http://www.domain.com/cgi-bin/anything.cgi
http://www.domain.com/cgi-bin/anything.pl
```

Apache intentará ejecutar estos archivos como programas CGI. Por supuesto, si estos archivos no son realmente ejecutables o no existen, Apache desplegará mensajes de registro de error.

Las asignaciones de permisos para el directorio del programa CGI, mencionadas antes, siguen aplicándose a esta configuración. La misma configuración se aplica también a los sitios de host virtuales. Por ejemplo, en el ejemplo siguiente, el alias /cgi-bin/ para www.client01.com se asigna para ejecutar programas CGI en el directorio /www/client01/public/cgi-bin:

```
<VirtualHost 192.168.2.100>

    ServerName www.client01.com
    DocumentRoot "/www/client01/public/htdocs"

    CustomLog logs/www.client01.com.access.log
    ErrorLog logs/www.client01.com.errors.log

    Alias /cgi-bin/ "/www/client01/public/cgi-bin/"

    <Directory "/www/client01/public/cgi-bin">
        Options ExecCGI -Indexes
        AddHandler cgi-script .cgi .pl
    </Directory>

</VirtualHost>
```

Aquí, el alias /cgi-bin/ para www.client01.com se asigna para ejecutar programas CGI en el directorio /www/client01/public/cgi-bin.

## Permitir el acceso cgi-bin a sus usuarios

Muchos Internet Service Providers (ISP) ofrecen espacios para cuentas de correo en su sitio Web. Estos sitios Web tienen normalmente unas URL, del tipo:

<http://www.isp.net/~username>

Estos sitios a menudo obtienen solicitudes para acceso cgi-bin mediante sus clientes Web (navegadores). El término *acceso cgi-bin* es un término general que se utiliza para indicar facilidad CGI en un servidor Web. Tradicionalmente, el directorio del programa CGI tiene el alias /cgi-bin/, y por eso se creó este término. El otro término común que se está haciendo muy popular es *página home*, que se refiere a la página indexada en el máximo nivel de un directorio Web de un usuario. Las siguientes secciones discuten dos modos de proporcionar acceso cgi-bin a usuarios en un servidor Web Apache. Sólo tiene que implementar cualquiera de estos métodos.

### Contenedores Directory o DirectoryMatch

Cuando se asigna la directiva UserDir al nombre de un directorio, Apache lo considera como el directorio de máximo nivel para un usuario del sitio Web, por ejemplo:

```
ServerName www.domain.com
UserDir public_html
```

En este momento, cuando hay una solicitud para `http://www.domain.com/~username`, Apache localiza el directorio del usuario local (normalmente comprobando el archivo `/etc/passwd` de los sistemas Unix), y entonces adjunta el directorio específico de `UserDir` para crear el nombre de la ruta para el directorio Web del usuario de máximo nivel. Por ejemplo, la URL

```
http://www.domain.com/~joe
```

hace que Apache busque `/home/joe/public_html` (suponiendo que `/home/joe` es el directorio local de joe). Si existe el directorio, se enviará la página `index` de este directorio al cliente que está realizando la solicitud.

Un modo de añadir soporte CGI para cada usuario es añadir la configuración siguiente en el archivo `httpd.conf`:

```
<Directory ~ "/home/[a-z]+/public_html/cgi-bin">
    Options ExecCGI
    AddHandler cgi-script .cgi .pl
</Directory>
```

O puede utilizar esta configuración:

```
<DirectoryMatch "/home/[a-z]+/public_html/cgi-bin">
    Options ExecCGI
    AddHandler cgi-script .cgi .pl
</DirectoryMatch>
```

En ambos métodos, Apache traduce solicitudes `http://www.yourcompany.com/~username/cgi-bin/` a `/home/username/public_html/cgi-bin/` y permite que se ejecute cualquier programa CGI con la extensión apropiada (`.cgi` o `.pl`).

**NOTA:** Todos los nombres de usuario deben estar escritos con letras minúsculas para que esto funcione. Si tiene nombres de usuario que son alfanuméricos, tiene que utilizar una expresión regular distinta. Por ejemplo, si tiene nombres de usuarios como `steve01` o `steve02`, tiene que cambiar el conjunto de caracteres `[a-z]+` para incluir los números, utilizando `[a-zA-Z0-9]+`. Si además permite nombres de usuarios con letras mayúsculas, tiene que utilizar la expresión regular `[a-zA-Z0-9]+`.

## ScriptAliasMatch

Utilizando la directiva `ScriptAliasMatch`, puede soportar directorios de programas CGI para cada usuario. Por ejemplo:

```
ScriptAliasMatch ^~([a-z]+)/cgi-bin/(.*) /home/$1/public_html/
cgi-bin/$2
```

Hace corresponder `username` con la variable de referencia `$1`, en la que `$1` es igual a `~username` y en la que `username` es un string con todas las letras minúsculas, como `joe` o `steven`, y entonces Apache hace corresponder a todo lo que va seguido por `/cgi-bin/` con la variable de referencia `$2`, por lo que Apache utiliza las variables `$1` y `$2` para crear la verdadera ruta del programa CGI. Por ejemplo:

```
http://www.domain.com/~joe/cgi-bin/search.cgi?author=kabir
```

Aquí `([a-z]+)` integrará una o más letras minúsculas seguidas de una señal `(~)` a `$1`. En otras palabras, esta expresión regular. Por eso, `$1` está asignada `joe` en el ejemplo anterior. Observe que `^` asegura que la directiva sólo se aplica a las URL que comiencen con `~`, tal y como muestra el ejemplo anterior.

La siguiente expresión regular en la directiva es `(.*)`, la cual integra todo lo que sigue el `/cgi-bin/` con `$2`. Por esos, `$2` está asignada a `search.cgi?author =kabir`. Ahora Apache puede crear la ruta física del directorio del programa CGI, utilizando:

```
/home/$1/public_html/cgi-bin/$2
```

Esta expresión regular da lugar a la siguiente ruta:

```
/home/joe/public_html/cgi-bin/search.cgi?author=kabir
```

Como el programa CGI `search.cgi` se guarda aquí, este programa se ejecuta y devuelve resultados al cliente Web.

Si no le gusta tener el directorio del programa CGI bajo `public_html` (es decir, el directorio específico de `UserDir`), puede mantenerlo fuera de este directorio eliminando la parte `public_html` de la expresión del siguiente modo:

```
ScriptAliasMatch ^~([a-z]+)/cgi-bin/(.*) /home/$1/cgi-bin/$2
```

Esto integrará con la siguiente solicitud URL:

```
http://www.domain.com/~joe/cgi-bin/search.cgi?author=kabir
```

en el siguiente archivo físico:

```
/home/joe/cgi-bin/search.cgi?author=kabir
```

Por supuesto, si no está de acuerdo con mantener un subdirectorío de usuario al que todo el mundo tenga acceso (es decir, `public_html`), puede remediarlo creando una partición Web (o un directorio) para sus usuarios y darles directorios individuales para que alojen sus páginas home. A continuación tenemos un ejemplo:

```
ScriptAliasMatch ^~([a-z]+)/cgi-bin/(.*) /www/$1/cgi-bin/$2
```

Esto hace corresponder solicitudes a `/www/username/cgi-bin/scriptname` y como este directorio no es el directorio del usuario `home` (`/`

home/username), debería, como administrador del sistema, ser capaz de ejercer un mayor control sobre él.

## Crear nuevas extensiones CGI utilizando AddType

Si quiere crear extensiones CGI nuevas en un directorio en concreto, puede utilizar .htaccess (o el archivo especificado por la directiva AccessFileName).

Antes de que pueda añadir extensiones nuevas utilizando el archivo de control de acceso a directorios (.htaccess), tiene que crear un contenedor <Directory> del siguiente modo:

```
<Directory "/path/to/your/directory">
    Options ExecCGI -Indexes
    AllowOverride FileInfo
</Directory>
```

La primera directiva dentro del contenedor de directorio, le dice a Apache que quiere permitir la ejecución del programa CGI en este directorio y desactivar la lista de características del directorio por razones de seguridad. La segunda directiva le dice a Apache que permita la característica FileInfo en el archivo de control de acceso a directorios (.htaccess). Esta característica le permite utilizar la directiva AddType en el archivo de control de acceso a directorios.

Para añadir una nueva extensión al programa CGI (.wizard), todo lo que necesita hacer es crear un archivo .htaccess (o cualquier otro que tenga especificado en la directiva AccessFileName) en el directorio:

```
AddType application/x-httplib-cgi .wizard
```

Entonces, vuelva a nombrar un programa CGI que exista ya en ese directorio para tener la extensión .wizard, y solicítelo con su navegador. Asegúrese de que todas las asignaciones de permisos a los archivos para el directorio y el programa CGI se refieren a la lectura y a la ejecución por Apache.

## Ejecutar programas CGI

Existen muchas posibilidades, en el caso de que sea un administrador de Apache, que tenga que establecer programas CGI, o incluso que tenga que saber escribirlos. En esta sección, veremos los conceptos básicos de la creación de programas CGI muy sencillos. Como este libro no es un libro de programación en CGI, no voy a proporcionar conocimientos con detenimiento. Mi objetivo es comentar determinados aspectos sobre la programación en CGI, que ayudarán a los administradores de Apache a gestionar mucho mejor sus sitios Web con CGI.

Muchos de los ejemplos de esta sección, utilizan Perl. Si no tiene Perl en su sistema, puede obtener la fuente, o posiblemente los binarios, de [www.perl.com](http://www.perl.com) que ha creado otra persona.

## Escribir scripts CGI en Perl

Un script CGI es sinónimo a un script Perl; Perl es el rey de todos los lenguajes de script, y fue el lenguaje Perl el que le dio fama a los scripts CGI. He escrito scripts CGI basados en Perl desde 1995, y continuo utilizando Perl para soluciones Web pequeñas y medianas. Como este libro no es un libro de programación en Perl, no vamos a introducirnos en la escritura de scripts de Perl en general. Si no está familiarizado con Perl, le recomiendo que lea un libro de programación de Perl tan pronto como pueda.

Cuando escribimos scripts CGI en Perl, es necesario seguir el siguiente conjunto de guías o estilo de programación:

- **Separar contenido y lógica; mantenga el contenido fuera de los scripts.** Utilice plantillas HTML o XML para asegurar que el aspecto de la interfaz de su script CGI no sea el propio script. Esto asegurará que un usuario que no sea programador, como un experto en gráficos o en HTML, pueda cambiar fácilmente esta interfaz.
- **Utilice archivos de configuración; nunca utilice información codificada por hardware en un script.** Utilice un archivo de configuración para leer la información. Esto permite que sus scripts sean más flexibles.
- **Normalice los datos de usuario; cuando recopile datos de los usuarios para usos futuros, asegúrese de normalizar los datos antes de almacenarlos en archivos o en una base de datos.** Por ejemplo, si recopila direcciones de correo electrónico de sus visitantes, sería una buena idea normalizar cada dirección de correo electrónico en su tipo de letra preferido (mayúsculas o minúsculas) y arreglar las omisiones de los usuarios y los errores de entrada. He visto realizar instancias en las que usuarios AOL escriben sus direcciones de correo electrónico en formularios Web sin el .com o poniendo espacios extra entre la arroba @ y el nombre o el nombre del host. También he visto miles de instancias en las que los usuarios introducen su nombre mezclando mayúsculas con minúsculas. Simplemente piense en la impresión que dejaría en ellos si personaliza un correo con un nombre no normalizado (tal y como lo introduce el usuario) o cuántos rechazos obtendrá cuando su sistema de listas de correo encuentre miles de errores del tipo "@AOL" o "user @ aol".
- **Comprobación de los datos del usuario; los scripts CGI a menudo constituyen el objetivo de muchos ataques contra la seguridad.** Si su script CGI acepta una entrada de un usuario, tiene que validar el dato del usuario antes de utilizarlo.

**TRUCO:** Antes de que empiece a hacer las cosas por su cuenta, diríjase al sitio de Comprehensive Perl Archive Network (CPAN) en <http://cpan.perl.com> para saber cuáles son los módulos Perl que puede utilizar para resolver su problema actual o reducir sus esfuerzos de desarrollo reutilizando módulos CPAN. En esta sección, voy a utilizar muchos módulos CPAN para construir scripts CGI. Cada vez que vea un módulo en cualquiera de los scripts que se discuten aquí, puede añadir el módulo en su sistema utilizando el módulo CPAN que se distribuye con el estandar de Perl.

**NOTA:** Los usuarios de Apache para Windows deberían consultar la documentación de Perl para saber cómo tienen que utilizar los módulos CPAN, ya que es distinta que la aproximación estándar discutida aquí. Ver, también, la sección específica para Windows de este libro.

Por ejemplo, imagine que ve un módulo llamado `HTML::Template` en una lista de scripts y que le gustaría instalar este módulo para poder ejecutar el script que utiliza. Para instalar el módulo CPAN, ejecute `perl -MCPAN -e shell` desde el prompt de comandos como raíz.

**NOTA:** Si está ejecutando el comando `shell perl -MCPAN -e` por primera vez, le pedirá que configure el módulo `CPAN.pm` actual, el cual se utiliza para instalar otros módulos CPAN. Simplemente siga las instrucciones para configurar el módulo `CPAN.pm` y luego proceda a ejecutar el comando.

Una vez que se encuentra en el prompt de CPAN, ejecute el comando `install HTML::Template` para instalar el módulo. El módulo CPAN instalará este módulo para usted. Si reclama dependencia, debería instalar algún otro módulo antes de poder instalar un módulo que dependa de otros módulos CPAN. Una vez que ha llevado esto a cabo, ejecute `quit` desde el prompt CPAN para volver al shell.

**TRUCO:** Si no quiere ejecutar el shell CPAN interactivo utilizando el shell `perl -MCPAN -e` para instalar un módulo, puede ejecutar `perl -MCPAN -e 'CPAN::Shell->install(modulename)'`. Por ejemplo, `perl -MCPAN -e shell 'CPAN::Shell->install('HTML::Template')'` instalará el módulo `HTML::Template`. También puede ejecutar este comando desde el script `shell`.

## Análisis de un script CGI sencillo

Los scripts CGI se utilizan normalmente para tomar la entrada del usuario y realizar una o más operaciones basadas en los resultados de una página HTML. En esta sección le mostraré cómo crear un script sencillo y entender los conceptos de los scripts CGI. El script que desarrollamos aquí realizará una sola tarea: tomará el nombre completo del usuario, le dará el formato adecuado y devolverá un mensaje de bienvenida personalizado. El listado 12.2 muestra el script greetings.pl realizando esta tarea.

**Listado 12.2.** greetings.pl

```
#!/usr/bin/perl
#
#
# Nombre: greetings.pl
#
# Propósito:
#
# Mostrar un mensaje de bienvenida
#
#####
use strict;
use CGI;

#
# Obtener datos del usuario
#
my $query = new CGI;
my $name = $query->param('name');

#
# Procesa datos
#
$name      = lc($name);
my @strArray = ();

foreach my $str (split(/ /,$name)) {
    push(@strArray, ucfirst($str));
}

my $formattedName = join(' ', @strArray);

#
# Muestra resultados
#
print $query->header;
print $query->start_html('Greetings');
```

```
print $query->p( 'Hello ' . $formattedName . ',' );
print $query->p( 'Thanks for coming to our Web site.' );
print $query->end_html;

#
# Fin
#
exit 0;
```

Ahora, vamos a analizar este script con más detalle. La primera línea es una línea especial:

```
#!/usr/bin/perl
```

Esta línea le dice al sistema que ejecute Perl cada vez que se ejecute este script. Si tiene instalado Perl en un directorio no estándar, entonces debe modificar esta línea. Por ejemplo, si instaló Perl en /usr/local/bin/perl, entonces debe cambiar esta línea para reflejarlo. Todas las líneas (menos la primera) que empiezan con un signo '#' son comentarios y Perl las ignora.

El siguiente segmento de código es:

```
use strict;
use CGI;
```

El `use strict` es un pragma Perl (piense en un pragma como en una directiva), que le dice a Perl que evalúe el script en cuanto a prácticas de programación peligrosas. La siguiente línea le dice a Perl que cargue un conocido módulo, que no forma parte de la distribución estándar de Perl, el módulo `CGI.pm`, el cual facilita la escritura de programas CGI. Este módulo maneja todos los detalles importantes de la obtención de datos de entrada que el servidor Apache pasa al script mediante el `STDIN`, analiza y descodifica los campos introducidos, proporciona métodos para desplegar cabeceras de contenido, crea contenido HTML, y realiza, en general, este tipo de tareas. Se trata de un super módulo disponible para usted. No se deben escribir scripts CGI en Perl sin el módulo `CGI.pm`. La única excepción a esta regla es en el caso de que tenga una restricción de recursos y no quiera cargar muchos módulos cada vez que se llama a un script. Pero en la mayoría de los casos, esto no es así, especialmente debido a que no se recomiendan soluciones CGI para sitios Web con alta carga.

El siguiente segmento de código es:

```
my $query = new CGI;
my $name = $query->param('name');
```

Aquí se crea un nuevo objeto CGI llamado `$query` y se llama al método `param()` del objeto CGI, el `$query`, para obtener el valor 'name' de la entrada del usuario. El valor de esta entrada se almacena en la variable `$name`. Por ejemplo, si se llama al script `greetings.pl` del siguiente modo:

```
http://www.domain.com/cgi-bin/greetings.pl?name=kabir
```

entonces el objeto CG, recibirá name=kabir como la entrada de usuario desde el servidor Apache y el script será capaz de acceder al valor ('kabir') mediante una llamada al método \$query->param('name').

El siguiente segmento de código es el meollo del script; es el bloque de procesamiento:

```
$name      = lc($name);
my @strArray = ();

foreach my $str (split(/ /,$name)) {
    push(@strArray, ucfirst($str));
}

my $formattedName = join(' ', @strArray);
```

Aquí se procesa la variable \$name. La primera línea utiliza la función de Perl built-in lc(), que transforma el string a letras minúsculas. Como lc() presenta la cadena \$name como el parámetro, devuelve un valor en letras minúsculas del nombre, que almacenamos en el nombre de la variable \$name. Hemos transformado todo el string en minúsculas pero podemos dejar únicamente la primera letra de cada nombre en mayúscula. Ahora un usuario llamado Carol Godsave podría introducir varias combinaciones distintas de su nombre. Por ejemplo:

```
Carol Godsave
CAROL GODSAVE
carol godsave
carol GODSAVE
CAROL godSave
carol GoDsAvE
```

Como puede ver, el usuario puede introducir una de estas u otras combinaciones de letras para representar su nombre. Sin embargo, queremos mostrar "Carol Godsave" ya que ese es el formato correcto para su nombre. Nuestro bloque de código de procesamiento utiliza el siguiente algoritmo:

1. Transforma el nombre en letras minúsculas. Esto convierte lo que Carol introduzca en 'carol godsave'.
2. Separa las palabras con un carácter espacio. Esto nos produce 'carol' y 'godsave'.
3. Convierte en letra mayúscula únicamente la primera letra de cada palabra. Esto produce 'Carol' y 'Godsave'.
4. Une las partes separadas por un espacio. Esto devuelve 'Carol Godsave', que es exactamente lo que queremos.

En este segmento de código utilizamos un array llamado @strArray, que iniciamos con una lista vacía. Entonces se utiliza un loop foreach, que podemos encontrar:

trar en Perl, para recorrer cada parte del nombre que se ha separado utilizando la función `split()`. Llamamos a la función `split()` con dos parámetros: `separator`, que, en este caso, es un espacio escrito en formato de expresión regular `//`, y la cadena que vamos a separar, que en este caso es `$name`. La función `split()` devuelve las partes separadas en un array. De modo que, efectivamente, en el caso de Carol, el loop `foreach` es del siguiente modo:

```
foreach my $str ('carol' 'godslove') {  
    push(@strArray, ucfirst($str));  
}
```

El loop recorre cada parte y cambia a mayúsculas únicamente el primer carácter de cada palabra utilizando una llamada a la función built-in `ucfirst()`. Cuando `ucfirst()` recibe un string, sólo convierte la primera letra a mayúscula y devuelve la cadena modificada. Por ejemplo, `ucfirst('carol')` devuelve 'Carol'; de igual modo, `ucfirst('godslove')` devuelve 'Godslove'. Cada parte modificada del nombre, se almacena en un array llamado `@strArray`. Por ejemplo, después de procesar el nombre de Carol, `@strArray` es `array = ('Carol', 'Godslove')`. Para terminar, unimos las partes del nombre en una nueva variable llamada `$formattedName` para formar el nombre completo. La función `join()` se utiliza para unir los elementos de `@strArray` con un espacio en blanco en la llamada `join(' ', @strArray)`. Ahora tenemos formado el nombre en la variable `$formattedName` y simplemente tenemos que mostrarlo. El segmento de código para mostrarlo es el siguiente:

```
print $query->header;  
print $query->start_html('Greetings');  
print $query->p( 'Hello ' . $formattedName . ',' );  
print $query->p( 'Thanks for coming to our Web site.' );  
print $query->end_html;
```

La primera línea de este segmento asigna la cabecera de contenido. Se necesita una cabecera `Content-Type`. El cliente utiliza esta cabecera para determinar cómo se despliega el contenido enviado por el servidor. Como un script CGI genera el contenido, debe decirle al servidor qué tipo de contenido está pasando al servidor para enviárselo al cliente. El módulo CGI proporciona un método llamado `header()`, que crea la cabecera `Content-Type` apropiada. Por ejemplo, `$query->header();` devuelve `Content-Type: text/html; charset=ISO-8859-1`. Cuando llamamos a `header()` sin un tipo específico de parámetro, el `Content-Type` por defecto será `text/html`. Sin embargo, si su script tiene que sacar un tipo distinto, por ejemplo `text/plain`, puede utilizar el parámetro `header(-type => 'text/plain')`. Por ejemplo:

```
#!/usr/bin/perl  
  
use CGI;
```

```

my $query = new CGI;

print $query->header(-type=>'image/gif');

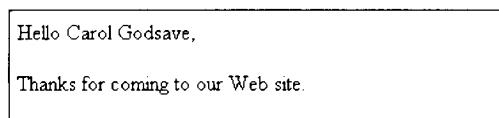
open(GIF, "/tmp/weather.gif");
while(<GIF>){ print; }
close(GIF);

```

Este script muestra un archivo GIF llamado /tmp/weather.gif utilizando la cabecera Content-Type: image/gif. Ahora, volvemos al script greetings.pl. Una vez que hemos desplegado la cabecera de contenido por defecto, el script utiliza métodos HTML del módulo CGI, como son start\_html(), p() y end\_html(), para crear el contenido HTML. Por ejemplo, imagine que Carol introduce la siguiente URL:

```
http://server/cgi-bin/greetings.pl?name=carol godsavE
```

Su navegador Web codificará automáticamente el espacio entre carol y godsavE en %20 (un número hexadecimal 20, que es equivalente al decimal 32, que es el carácter espacio en el código ASCII). El módulo CGI descodifica esto y produce el script 'carol godsavE' como nombre. El script muestra una página como la que se muestra en la figura 12.4.



**Figura 12.4.** Salida del script greetings.pl

El método p() crea un par de etiquetas de párrafo HTML, que se utilizan para meter el parámetro que recibe. Por ejemplo, la primera llamada p():

```
print $query->p( 'Hello ' . $formattedName . ',' );
```

traduce a:

```
<p>Hello Carol GodsavE,</p>
```

El documento completo de salida que envía el navegador Web es del siguiente tipo:

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
    "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
      lang="en-US">

<head>

```

```
<title>Greetings</title>
</head>
<body>

<p>Hello Carol Godsave,</p>
<p>Thanks for coming to our Web site.</p>

</body></html>
```

El método `start_html()` produce el contenido siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
    "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
      lang="en-US">

<head>
    <title>Greetings</title>
</head>
<body>
```

Observe el título 'Greetings' que se pasó al método. Los dos métodos `p()` producen lo siguiente:

```
<p>Hello Carol Godsave,</p>
<p>Thanks for coming to our Web site.</p>
```

Y finalmente, el método `end_html()` produce:

```
</body></html>
```

que finaliza el documento HTML.

Si realiza una lectura cuidadosa observará que hemos violado la primera regla de estilo de la programación CGI, que dicta que deberíamos separar el contenido y la lógica. Este pequeño script tiene HTML dentro del script, lo que hace más difícil su modificación por parte de alguien que no sea programador. Pero este script era simplemente un ejercicio. En la siguiente sección, veremos un script de gran utilidad, en grado de producción, que cumple las líneas de estilo marcadas.

## Crear un procesador básico de formularios Web

Los scripts CGI comunes se utilizan como procesos especializados del lado del servidor de un formulario Web. Virtualmente, cada sitio Web tiene un formulario Web que toma las entradas del usuario y las almacena en un archivo o en una base de datos.

En la siguiente sección, veremos un script CGI muy adaptable que puede funcionar virtualmente con cualquier formulario Web.

## **Características del procesamiento básico de un formulario Web con un script CGI**

Imagine que su sitio Web tiene uno o más formularios Web que necesitan recopilar datos de los visitantes Web para almacenarlos en archivos o en bases de datos. Una solución basada en un script CGI, tendrá las siguientes características:

- **Soporte de varios formularios Web:** un solo script funcionará para varios formularios Web de una página. Un formulario Web de una sola página hace todas sus preguntas en una sola página. Hay veces en las que le pedirán que construya un formulario Web de varias páginas. Estas aplicaciones de procesamiento de formularios normalmente utilizan lógica de negocio personalizada para desplegar varios formularios y se encuentran fuera del alcance de este libro. Mi recomendación personal, basada en años de desarrollo Web, es evitar formularios de varias páginas porque la gente a menudo se molesta cuando les piden que contesten a una serie muy larga de preguntas. Además, cuando diseñamos un formulario Web de una sola página, no debemos realizar demasiadas preguntas. Mantener la colección de datos en mínimos es una buena práctica. La experiencia me dice que las esperanzas e ilusiones se terminan con los formularios de gran tamaño o de varias páginas, y a menudo deciden abandonar el posible negocio con el sitio.
- **Configuración centralizada para todos los formularios Web:** el script utiliza un archivo de configuración que soporta varios formularios Web. Cada configuración de un formulario Web puede almacenarse por separado en este archivo de configuración estándar, lo que significa que puede centralizar la información sobre la configuración de formularios, lo cual es muy ventajoso para un administrador de sistemas muy ocupado.
- **Archivos con los valores separados por comas (Comma Separated Value, CSV) para almacenar los datos:** el script le permite guardar datos en archivos CSV. Cada formulario puede tener su propio archivo CSV. Además puede especificar el orden de los campos y qué archivos quiere almacenar.

Voy a suponer que tiene un formulario de registro llamado `register.html`, como el que se muestra en el listado 12.3

### **Listado 12.3. register.html**

```
<html>  
<body bgcolor="white">  
<font face="Arial" size=+1>User Registration Form</font><br>
```

```

<p>
<form action="/cgi-bin/formwizard.pl" method="POST">

<table border=0
       cellpadding=3
       cellspacing=0
       bgcolor="#000000">
<tr>
<td>

<table border=0
       cellpadding=5
       cellspacing=5
       bgcolor="#abcdef">

<tr>
<td> Name </td>
<td> <input name="name" type="text" size=30 maxsize=50> </td>
</tr>

<tr>
<td> Email </td>
<td> <input name="email" type="text" size=30 maxsize=50> </td>
</tr>

<tr>
<td> Zipcode</td>
<td> <input name="zipcode" type="text" size=30 maxsize=50> </td>
</tr>

<tr>
<td> Can we send you junk mail?</td>
<td>
<input type=radio name="opt-in" value="yes">Yes, Please!
<input type=radio name="opt-in" value="no">No, Never!
</td>
</tr>

<tr>
<td align=center> <input type=submit value="Register Me"></td>
<td align=center> <input type=reset> </td>
</tr>

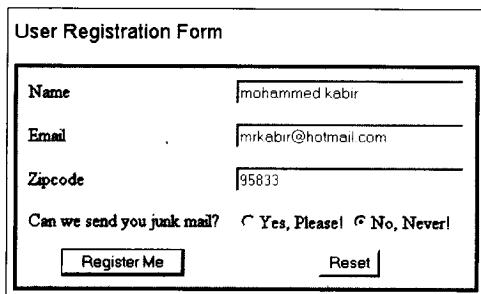
</table>

</td>
</tr>
</table>
</form>
<p>

</html>

```

Podemos ver este formulario Web en la figura 12.5.

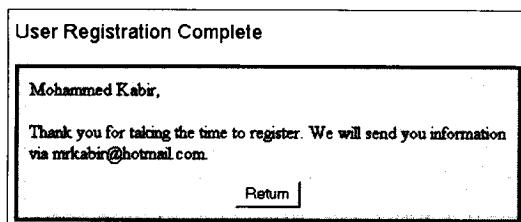


User Registration Form

Name	mohammed kabir
Email	mrkabir@hotmail.com
Zipcode	95833
Can we send you junk mail? <input type="radio"/> Yes, Please! <input checked="" type="radio"/> No, Never!	
<input type="button" value="Register Me"/>	<input type="button" value="Reset"/>

Figura 12.5. El formulario register.html en un navegador Web

Este formulario Web tiene cuatro campos de datos: name (nombre), email (dirección de correo electrónico), zipcode (código postal) y una casilla de conformidad. Este es el típico formulario de registro de usuarios excepto en que la mayoría de los sitios no tienen una casilla de conformidad como la que aparece en este formulario. Cuando un usuario rellena este formulario, el dato se almacena en un archivo, y el usuario recibe un mensaje de agradecimiento o se le dirige a otra página. Por ejemplo, cuando se rellena el formulario anterior, puede mostrar un mensaje de agradecimiento como el que se muestra en la figura 12.6.



User Registration Complete

Mohammed Kabir,

Thank you for taking the time to register. We will send you information via [mrkabir@hotmail.com](mailto:mrkabir@hotmail.com).

Figura 12.6. Mensaje de agradecimiento por llenar el formulario register.html

## Desarrollar un script CGI en Perl para procesar un formulario Web

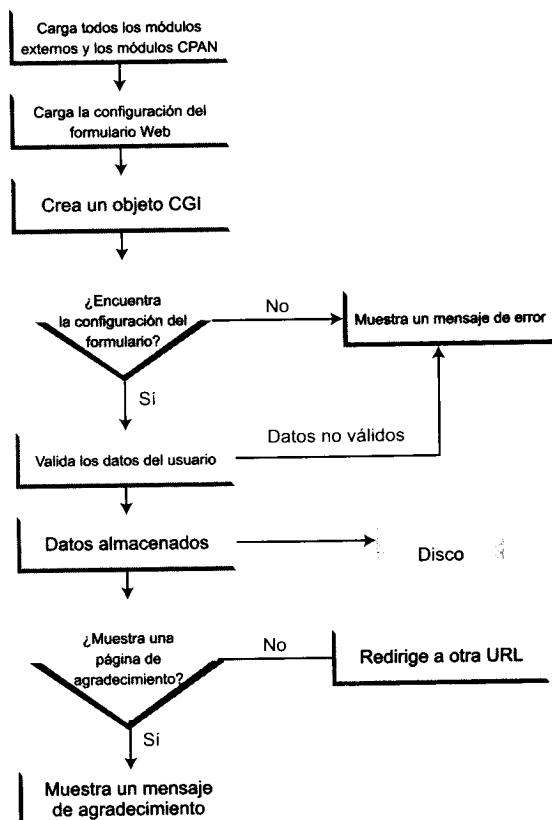
A continuación vamos a desarrollar un script CGI en Perl que le permite crear una solución de procesamiento de un formulario Web tal y como muestra en el ejemplo de la última sección. La figura 12.7 muestra un diagrama de flujo de este tipo de solución CGI. Cuando el formulario llama al script register.html:

```
<form action="/cgi-bin/formwizard.pl" method="POST">
```

Este script realiza las tareas siguientes:

1. Carga todos los estándares externos (por ejemplo, parte de la distribución estándar de Perl) y otros módulos CPAN.
2. Carga el archivo de configuración central, que debería tener una configuración específica para el formulario Web al que adjuntan datos.

3. Crea un objeto CGI que le permite recuperar datos adjuntos y acceder a otra información suministrada por el servidor Web.
4. Determina si el formulario Web que activó el script, tiene una configuración en el archivo de configuración central. Si no la tiene, entonces simplemente muestra un mensaje de error indicando que este formulario no se puede procesar con este script. Si la tiene, comienza el proceso.
5. Como en primer paso del proceso, el script valida el dato del usuario. Este paso implica la comprobación de los datos basada en los requisitos indicados en el archivo de configuración y en la realización de comprobaciones de seguridad de los campos suministrados por el usuario.
6. Si el paso de validación no devuelve un error, el dato se almacena en un archivo de datos. Por el contrario, si falla el paso de validación debido a un error del usuario, entonces se muestra un mensaje de error específico para el tipo de error y se abandona el proceso.
7. Si el paso anterior tiene éxito, el script muestra un mensaje de agradecimiento o redirige al usuario a otra URL y termina.



**Figura 12.7.** Diagrama de procesamiento CGI de un formulario Web

## El script formwizard.pl

El script formwizard.pl implementa este diagrama de flujo. En el archivo de configuración formwizard.ini, el campo register.html tiene las siguientes restricciones:

- Es necesario el campo name (nombre) y tiene asignada una longitud de 3 a 50 caracteres. El valor del campo debe estar compuesto por caracteres comprendidos de la a- a la z, y por el punto. Si el valor del campo pasa las pruebas de longitud y de patrón, se almacena como una cadena en la que cada palabra comienza con una letra mayúscula.
- También es necesario el campo email (dirección de correo electrónico), y debe tener entre 6 y 100 caracteres. Los caracteres aceptables para una dirección de correo electrónico son de la a a la z, del 0 al 9, el guión (-), el guión bajo (\_), el punto, y la arroba (@). La prueba de patrón de email se aplica al valor de este campo y si el valor pasa todas las pruebas (longitud, conjunto de caracteres, y pruebas de patrón), se guarda en formato en letras minúsculas.
- El campo zipcode (código postal) es un campo opcional de números, que debe pasar la prueba de patrón us\_zipcode si el usuario introduce un valor. La prueba us\_zipcode simplemente comprueba si el número introducido es un número de cinco dígitos o un número de cuatro dígitos para formar códigos postales del tipo 95825-1234. Los únicos caracteres permitidos son del 0 al 9 y el guión.
- Finalmente, el campo opt-in se considera un campo de texto necesario que únicamente puede tener los valores 'yes' o 'no'.

**CD-ROM:** El script formwizard.pl está incluido en el CDROM.

**Listado 12.4.** Archivo de configuración formwizard.ini

```
# -----
# 
# Configuración del formulario Web para *** register.html ***
# 
# -----
[register.html]
template_dir = /www/asb2/ch12/forms/register

# -----
# Tras procesar el formulario, el script debe comunicar al
# usuario que sistema recibió el dato. Esto se realiza utilizando
# un mensaje "Thank You" (que llamaremos thankyou)
```

```

#
# Asigna el nombre de la plantilla thank you
#
thankyou_template = thanks.html

#
# Si quiere redirigir al usuario a otra URL,
# asigne la URL aquí. No lo asigne a ninguna URL
# si quiere mostrar el mensaje thankyou
#
# Si se asigna la URL, se utiliza siempre en lugar de
# mostrar el mensaje thankyou.
#
thankyou_redirect_url = /asb2/ch12/forms/feedback.html

#
# El script mostrará el mensaje thankyou o
# redirigirá al usuario a otra URL.
# Si desea enviar datos a la URL mediante el método GET
# fíjelo con el valor 'yes'; sino, asigne el valor 'no'.
#
send_data_on_redirect = no

#
# Tener una lista de campos del formulario
# asegura que sólo utilizamos lo que necesitamos.
# Ignoraremos cualquier campo o campos adicionales
# que introduzca algún malintencionado.
#
form_field_info=<<FORM_FIELDS
name,text,required,max_chars=50,min_chars=3,case=ucfirst,
exclude_regex=[^a-z .],pattern=none
email,text,required,max=100,min=6,case=lowerall,
exclude_regex=[^a-z0-9-_@],pattern=email
zipcode,number,optional,pattern=us_zipcode,exclude_regex=[^0-9-]
opt-in,text,required,case=lower,pattern=none,valid_data=yes|no
FORM_FIELDS

#
# Cuando tiene lugar un error de entrada, se muestra un mensaje.
# El mensaje tiene una lista del campo o de los campos que han
# desaparecido o que no son válidos.
# Puede personalizar el ACTION del mensaje. En otras palabras,
# introducir texto que le diga al usuario que
# realice correcciones y reenvíos.
#
data_error_msg = Please enter missing data and resubmit.

#
#
# Especifica el
data_directory=/tmp
data_filename=registration.csv

```

```
data_field_separator=,
data_field_order=email,name,zipcode,opt-in
```

## Análisis de formwizard.ini

A continuación, vamos a analizar este script más detenidamente, de modo que pueda configurarlo para sus necesidades específicas.

La primera línea es una línea especial:

```
#!/usr/bin/perl
```

Esta línea le dice al sistema que ejecute Perl cada vez que el sistema ejecute este script. Si tiene instalado Perl en un directorio no estándar, debe cambiarlo adecuadamente. Por ejemplo, si tiene instalado Perl en /usr/local/bin/perl, entonces cambie esta línea para reflejarlo.

El siguiente segmento de código es:

```
use strict;
use Config::IniFiles;
use File::Basename;
use HTML::Template;
use Fcntl qw(:DEFAULT :flock);
use CGI;
```

Esto carga todos los estándares y todos los módulos CPAN necesarios. Los módulos CPAN que utiliza este script son Config::IniFiles, HTML::Template y CGI. El módulo CGI se carga como un módulo Perl, pero debe instalar los otros dos utilizando el módulo CPAN. A continuación tenemos dos comandos quick que instalarán estos módulos desde la raíz shell:

```
perl -MCPAN -e 'CPAN::Shell->install('Config::IniFiles')
perl -MCPAN -e 'CPAN::Shell->install('HTML::Template')
```

También puede ejecutar:

```
perl -MCPAN -e 'CPAN::Shell->install('CGI')
```

para asegurarse de que su módulo CGI.pm es la última versión desde la red CPAN.

El siguiente segmento de código es:

```
my $cfg      = new Config::IniFiles -file => "formwizard.ini";
my $query    = new CGI;
my $formName = get_form_name($ENV{HTTP_REFERER});
```

Este segmento crea un objeto de configuración llamado \$cfg, que carga el archivo formwizard.ini. El archivo formwizard.ini es el archivo de configuración central para este script y debe residir en el mismo directorio que el script CGI, formwizard.pl. Si quiere mantenerlo en un directorio distinto al que se encuentra el script CGI, tiene que cambiar la ruta. Por ejemplo, para

almacenar el archivo de configuración en un directorio llamado /www/myscripts/conf, modifique el script del siguiente modo:

```
my $cfg      = new Config::IniFiles  
    -file => "/www/myscripts/conf/formwizard.ini";
```

asegúrese de que el usuario Apache (es decir, el usuario utilizado para la directiva User, en el archivo httpd.conf) puede leer el archivo de configuración. El script creado se llama \$query. A continuación, el script llama a una subrutina llamada get\_form\_name() para obtener el nombre del formulario Web que le llama. La subrutina es:

```
sub get_form_name {  
    my $referrer = shift;  
  
    # Si HTTP tiene una cadena de consulta  
    # devuelve sólo la parte de la cadena que no es de consulta  
    #  
    if ($referrer =~ /\?/) {  
  
        ($referrer, undef) = split(/\?/, $referrer);  
  
    }  
  
    return basename($referrer);  
}
```

Esta subrutina pasa desde la variable de entorno CGI HTTP\_REFERER cuando se la llama. Esta variable está asignada a la URL, que llama al script. Por ejemplo, si se puede acceder al formulario register.html desde una URL como http://rhat.nitec.com/asp2/ch12/forms/register.html, entonces cuando se envía al script /c/s.dll/formwizard.pl, el objeto CGI \$query determina que HTTP\_REFERER sea http://rhat.nitec.com/asp2/ch12/forms/register.html. El código get\_form\_name() devuelve la porción 'register.html' de la URL.

Entonces el script almacena el nombre del formulario Web en una variable global llamada \$formName.

El segmento de código siguiente es:

```
my $templateDir = $cfg->val($formName, 'template_dir');  
my @fieldInfo = $cfg->val($formName, 'form_field_info');
```

Este segmento pasa el nombre del formulario Web al objeto de configuración como un nombre de sección, y recupera información de configuración sobre el directorio de la plantilla actual del formulario Web, e información sobre los campos del formulario. La información se almacena en los arrays \$templateDir y @fieldInfo, respectivamente. El ejemplo de archivo de configuración, formwizard.ini, se muestra en el listado 12.4, que muestra una configura-

ción para un formulario Web llamado register.html. El formato del archivo de configuración es muy simple:

```
[section1]
key1 = value
key2 = value
key3 = value

key4 = <<MULTIPLE_VALUES
value1
value2
value3
...
valueN
MULTIPLE_VALUES
...
keyN = value
[section2]
```

Este tipo de archivos de configuración es común en el mundo de Windows y normalmente nos referimos a ellos como archivos "ini". Por eso, el módulo que hemos utilizado aquí se llama Config:IniFiles. En nuestro archivo formwizard.ini, el nombre de la sección es en realidad el nombre del formulario Web register.html.

La variable \$templateDir y el array @fieldInfo están asignados desde estas líneas de configuración:

```
[register.html]

template_dir = /www/asb2/ch12/forms/register

form_field_info=<<FORM_FIELDS
name,text,required,max_chars=50,min_chars=3,case=ucfirst,
exclude_regex=[^a-z .],pattern=none
email,text,required,max=100,min=6,case=lowerall,
exclude_regex=[^a-z0-9-_@],pattern=email
zipcode,number,optional,pattern=us_zipcode,exclude_regex=[^0-9-]
opt-in,text,required,case=lower,pattern=none,valid_data=yes|no
FORM_FIELDS
```

La línea template\_dir del archivo de configuración está asignada al directorio en el que se mantienen las plantillas HTML para este formulario Web (register.html). En la versión actual del script, la única plantilla que utilizamos es la plantilla HTML para la página thank you, de modo que se mantiene en este directorio.

La configuración form\_field\_info es una versión extendida del concepto del par clave=valor. Aquí una sola clave, form\_field\_info, tiene varios valores. Los valores son líneas entre las cadenas FORM\_FIELDS. Cada línea de valores tiene varios campos. Por ejemplo:

```
name, text, required, max_chars=50, min_chars=3, case=ucfirst,  
exclude_regex=[^a-z .], pattern=None
```

Aquí el nombre del campo del formulario se encuentra en register.html del siguiente modo:

```
<input name="name" type="text" size=30 maxsize=50>
```

Para controlar las entradas del usuario, el script utiliza un conjunto de parámetros para cada campo de entrada. Los parámetros se muestran en la tabla 12.1. Cada parámetro se separa con una coma y se almacena en el archivo de configuración del siguiente modo:

```
form_field_info=<<FORM_FIELDS  
fieldname1,parameter1,parameter2,parameter3,...,parameterN  
fieldname2,parameter1,parameter2,parameter3,...,parameterN  
fieldname3,parameter1,parameter2,parameter3,...,parameterN  
...  
fieldnameN,parameter1,parameter2,parameter3,...,parameterN  
FORM_FIELDS
```

**Tabla 12.1.** Parámetros de control de las entradas de usuario en los archivos de configuración

Parámetro	Ejemplo	Explicación
Text,	name,text	Este parámetro indica que el campo nombrado es una cadena de caracteres.
Number	zipcode,number	Este parámetro indica que el campo nombrado es un número.
required,	name,required	Este parámetro indica que el campo nombrado es un campo de entrada necesario. Si este campo no está, o no es válido, se mostrará un mensaje de error y se abandonará el procesamiento del formulario.
Optional	zipcode,optional	Este parámetro indica que el campo nombrado es opcional. Si este campo no está, continuará el proceso.
Max_chars=n	name,max_chars=50	Determina el número máximo de caracteres permitidos para el campo de entrada nombrado. Si el usuario introduce más caracteres de los especificados aquí, se mostrará un mensaje de error.
Min_chars=n,	name,min_chars=3,	Determina el mínimo número de caracteres permitidos en el campo de entrada nombrado. Si el usuario introduce menos caracteres del número especificado, se mostrará un mensaje de error.

Parámetro	Ejemplo	Explicación
case=ucfirst   upper   lower name,case=ucfirst   lcfirst		Antes de almacenar el campo de entrada nombrado, el script formatea el campo utilizando la función específica. Por ejemplo:  ucfirst: formatea el valor del campo del formulario para que tenga únicamente la primera letra mayúscula. Esto resulta útil cuando almacenamos nombres.  upper: formatea el valor del campo del formulario para que todas las letras sean mayúsculas.  lower: formatea el valor del campo del formulario para que todas las letras sean mayúsculas.  lcfirst: formatea el valor del campo del formulario para tener únicamente la primera letra en minúscula.
exclude_regex=[character set]	name,exclude_regex=[^a-z .]	Los únicos caracteres permitidos para el valor del campo del formulario nombrado. El resto están excluidos. Se genera un mensaje de error cuando el valor contiene caracteres excluidos.
pattern=none	email, pattern=email	El parámetro pattern determina un patrón especial para ser considerado en el campo del formulario nombrado. Si el valor del campo no corresponde al patrón especificado, se rechaza y se genera un mensaje de error.  Los patrones soportados son: none, email, us_zipcode y large_plain_text.
valid_data=value1 value2	opt-in, valid_data=yes no	Puede añadir una subrutina de comprobación de patrones llamada sub check_newpattern { # your checking code } para que el script soporte de nuevos patrones para otros tipos de campos de formularios.
	Aquí el campo opt-in puede tener únicamente el valor de 'yes' o 'no'.	Este parámetro se utiliza para los campos de texto que únicamente pueden tener uno de los valores especificados, que están separados por un carácter  .

Una vez que cargada la información de los campos del formulario en el array @fieldInfo, el siguiente segmento de código comprueba si el archivo de configuración tiene un directorio de plantillas (es decir, valores para la variable \$templateDir), o si el array @fieldInfo está vacío.

```

if ($templateDir eq '' || $#fieldInfo < 1 ) {
    print $query->header;
    print alert("Sorry, $formName is not managed by this
script.");
    exit 0;
}

```

Estas pruebas se realizan para determinar si el formulario está configurado en el archivo `formwizard.ini`. Si no está configurado, el script devuelve un mensaje de error indicando que el formulario no está gestionado por el script y finaliza el programa. Si el formulario Web se configura adecuadamente en el archivo `formwizard.ini`, entonces el script continúa al siguiente segmento de código:

```

my @errors = validate_data(\@fieldInfo);
if (@errors >= 0) {
    print $query->header;
    print alert(join("\n",@errors) . "\n\n" .
                 $cfg->val($formName,'data_error_msg'));
    exit 0;
}

```

Aquí el script llama a la subrutina `validate_data()`, que es pasada la referencia del array `@fieldInfo` para verificar que el dato introducido por el usuario no tiene errores. Esta subrutina lleva a cabo lo siguiente:

1. Recorre la información de cada campo del formulario encontrada en el array `@fieldInfo` (que se pasa a la subrutina como una referencia) y devuelve el campo name, el campo type, el campo requirements, y similares.
2. Para cada campo del formulario recupera el valor del objeto `$query` utilizando el método `$query->param($fieldname)`. El valor se almacena en la variable `$value`.
3. Si el campo es necesario y el valor está vacío, se almacena un mensaje de error en el array `@errors`, y continúa el loop con el siguiente campo. Si el campo es opcional y el usuario no ha introducido ningún valor, el loop continúa con el siguiente campo.
4. Cuando hay un valor (para un campo necesario u opcional), se llama a la subrutina `check_data_field()` para comprobar que el valor cumple los requisitos indicados en el archivo de configuración.
5. Si el valor del campo supera la comprobación de validación realizada por la subrutina `check_data_field()`, se cambia el tipo de letra dependiendo de los requisitos de la configuración. En otras palabras, el valor se pasa a mayúsculas, a minúsculas o a una mezcla y se almacena en el objeto CGI, utilizando el método `$query->param()`. La rutina

`check_data_field()` devuelve una lista de errores (en caso de existir).

6. Si la subrutina `check_data_field()` devuelve algún error, el script muestra un mensaje de error utilizando la subrutina `alert()` y abandona. Por otro lado, si no hay un informe de errores, el script continúa escribiendo datos en el disco utilizando la subrutina `write_data()`. `write_data()` utiliza el nombre del archivo de configuración. El dato del nombre completo de la ruta está basado en los parámetros de configuración `data_directory` y `data_filename`. Esta subrutina escribe datos introducidos por el usuario en el orden que aparecen en el parámetro de configuración `data_field_order`, utilizando el separador de campos en el parámetro `data_field_separator`. Observe que el archivo de datos está abierto solamente para adjuntar el modo y está bloqueado exclusivamente durante la operación de escritura de modo que no hay copias de `formwizard.pl` que tengan acceso durante este período crítico.
7. Entonces el script utiliza la opción de configuración `thankyou_redirect_url` que se encuentra en `formwizard.ini` para determinar si el usuario debería ser redirigido a otra URL o debemos mostrarle el mensaje de agradecimiento. Si `thankyou_redirect_url` no está vacío en el archivo de configuración, el script utiliza la subrutina `redirect_data()` para redirigir al usuario a dicha URL. La subrutina `redirect_data()` empaqueta todos los campos del formulario como pares clave=valor en la URL y redirige al navegador Web del usuario a la URL. Por ejemplo, si el formulario `register.html` se rellena con `name=mohammed kabir, zipcode=95833, email=MRKABIR@hotmail.com, y opt-in=yes`, y si el `thankyou_redirect_url` está asignado a `http://www.domain.com/friends.pl`, entonces la subrutina `redirect_data()` redirige el navegador Web a:

```
http://www.domain.com/
friends.pl?name=Mohammed%20Kabir&zipcode=95833&email=
mrkabir@hotmail.com&opt-in=yes
```

Esto le permite al sitio remoto recibir el dato del usuario mediante el método HTTP GET. Observe que si simplemente quiere redirigir al usuario a otra página y no enviar el dato mediante GET, tiene que asignar el valor de `no` al parámetro `send_data_on_redirect` en el archivo de configuración .

8. Si no está asignado el `thankyou_redirect_url`, el script carga el nombre de la plantilla de agradecimiento desde el archivo de configuración utilizando el parámetro de configuración `thankyou_template`. Se supone que esta plantilla se encuentra en el directorio de plantillas asig-

nadas por el parámetro de configuración `template_dir`. El script utiliza la subrutina `show_thanks()` para mostrar la página de agradecimiento. La subrutina `show_thanks()` personaliza el agradecimiento reemplazando etiquetas especiales con los valores que introduce el usuario en los campos. Esto se realiza creando una instancia del objeto `HTML::Template`, que se inicializa con la página de la plantilla asignada por el parámetro `thankyou_template`. El objeto `HTML::Template` llamado `$template` utiliza su método `its $template->param()` para reemplazar etiquetas especiales con un hash creado por el método `$query->Vars()`.

Para terminar, se muestra el resultado del script y este finaliza.

La plantilla de agradecimiento utilizada en este ejemplo se muestra en el listado 12.5.

**Listado 12.5.** Página `thankyou.html`

```
<html>

<body bgcolor="white">

<font face="Arial" size=+1>User Registration Complete</font><br>

<p>
<form action=<TMPL_VAR name=REFERRER>" method="GET">

<table border=0
       cellpadding=3
       cellspacing=0
       bgcolor="#000000">
<tr>
<td>

<table border=0
       cellpadding=5
       cellspacing=5
       bgcolor="#abcdef">

<tr>
<td>

<TMPL_VAR NAME=name>, <p>

Thank you for taking the time to register. We will send you
information via <TMPL_VAR NAME=email>.

</td>
</tr>
```

```

<tr>
<td align=center> <input type=submit value="Return"></td>
</tr>

</table>
</td>
</tr>
</table>
</form>
<p>

</html>

```

Observe que las etiquetas se reemplazan con los valores de los campos name y email que introduce el usuario. Recuerde que durante la validación de los datos, el script actualiza el tipo de letra de los campos que rellena el usuario, de modo que el valor que se muestra realmente, tiene el formato correcto, lo cual es perfecto cuando se muestra información de vuelta al usuario. Por eso, el usuario puede introducir carol godslove en el campo name pero la página de agradecimiento mostrará Carol Godsave, lo que gustará mucho al usuario. A continuación tenemos un ejemplo de un archivo de datos tras cuatro envíos de formulario register.html.

```

kabir@domain.com,Mohammed J. Kabir,12345,no
carol@domain.com,Carol Godsave,95833,yes
joegunchy007@aol.com,Joe Gunchy,07024,yes
jennygunchy007@aol.com,Jennifer Gunchy,07024,no

```

## Gestionar un formulario Web con formwizard.ini

Como ha podido observar en la sección anterior, el script formwizard.pl, que utiliza el archivo de configuración formwizard.ini, es muy configurable y le permite gestionar muchos formularios Web de una sola página utilizando un solo script y un solo archivo de configuración central. Para gestionar nuevos formularios Web, simplemente tiene que copiar el archivo de configuración en register.html y crear una nueva sección en el archivo formwizard.ini. A continuación tiene los pasos que necesita dar para gestionar un formulario Web llamado feedback.html utilizando este script.

1. Tiene que crear una sección llamada [feedback.html] en el archivo formwizard.ini:

```

[register.html]
template_dir      = /path/to/feedback/template/dir
thankyou_template = thanks.html
send_data_on_redirect = no

form_field_info=<<FORM_FIELDS

```

```
#  
# información sobre los campos  
#  
FORM_FIELDS  
  
data_error_msg = Please enter missing data and resubmit.  
  
data_directory=/feedback/data/directory  
data_filename=feedback.csv  
data_field_separator=,  
data_field_order=comma separated field list goes here
```

Realice los cambios necesarios observando la sección [register.html].

2. Cambie la línea action del formulario feedback a:

```
<form action="/c/s.dll/formwizard.pl" method="POST">
```

3. Tiene que crear la plantilla thanks.html en el directorio especificado por el parámetro template\_dir en la sección [feedback.html]. Este archivo de plantilla utilizará etiquetas <TMPL\_VAR name=fieldname> para personalizar el mensaje de agradecimiento.

Esto es todo lo que necesita para gestionar un nuevo formulario Web con formwizard.pl. La gestión de formularios Web es una gran parte de la carga de trabajo del CGI en la mayoría de los sitios Web, y el script formwizard.pl le ayuda a centralizar la gestión de los formularios Web, facilitándole el soporte de formularios.

**TRUCO:** Mientras desarrollaba el script, se lo pasé a mucha gente para que lo probase. Encontraban constantemente nuevas características y curiosos detalles para este script. He creado un sitio para un proyecto SourceForge para todos los scripts CGI que he creado bajo GNU Public License (GPL), de modo que puede obtener la última versión de este script, en cualquier momento, en <https://sourceforge.net/projects/mkweb/>.

## Crear un script que envíe una página por correo electrónico

Muchos sitios Web tienen una característica que permite al visitante enviar la página Web a un amigo o a un compañero mediante un correo electrónico. Este tipo de característica, permite al sitio Web aumentar su tráfico, y el departamento de marketing suele considerarla como una gran herramienta de marketing. En esta sección, le mostraré cómo puede añadir un script CGI a su sitio Web, que permita a sus visitantes enviar cualquiera de sus páginas Web a otra persona mediante un correo electrónico. Para empezar, vamos a ver cómo funciona.

Un usuario puede introducir la dirección de correo electrónico de un amigo o de un compañero y hacer clic en el botón de enviar para mandar esta página Web a alguien que pueda estar interesado en ella. El contenido de HTML no es importante; lo que importa es el pequeño formulario HTML embebido:

```
<form action="/cgi-bin/mime-mail.pl" method="POST">
<input type=text name="name" value="your-name" size=12
maxsize=50>&nbsp;<br>
<input type=text name="email" value="friend-email" size=12
maxsize=100>&nbsp;
<input type=submit value="Send">
</form>
```

Cuando el usuario introduce su nombre y la dirección del amigo o del compañero y hace clic en el botón de enviar, se llama al script /c/s.dll/mime-mail.pl, que se muestra en el listado 12.6.

#### Listado 12.6. /cgi-bin/mime-mail.pl

```
#!/usr/bin/perl -w
#
#
# Nombre: mime-mail.pl
#
# Propósito:
#
# enviar la página por correo electrónico
#
#####
use strict;

use CGI;
use MIME::Lite;
use MIME::Lite::HTML;
use Config::IniFiles;
use File::Basename;

#
# Crea un objeto CGI
#
my $query = new CGI;

#
# Crea un objeto de configuración cargando el archivo mime-mail.ini
# desde el directorio físico asignado en /cgi-bin/
#
my $cfg      = new Config::IniFiles -file => "mime-mail.ini";
#
#
```

```

# Obtiene la URL
#
my $referrerURL = $ENV{HTTP_REFERER};

#
# Obtiene el nombre de la página
my $pageName = get_form_name($referrerURL);

#
# Imprime la cabecera del content type para que sea text/html
#
print $query->header;

#
# Si la página no tiene una [section] en el archivo
# mime-mail.ini, utiliza la sección por defecto
$pageName = 'defaults' if ($cfg->val($pageName, 'from') eq '');

#
# Obtiene la dirección de correo electrónico. Si no se suministra,
# muestra un mensaje de error
#
my $to = ( $query->param('email') eq '' ||
           ! check_email($query->param('email'), 'email')) ?
           abort('Email address is missing.') : $query-
>param('email');

my $senderName = ( $query->param('name') eq '' ) ?
           abort('Your name is missing.') : $query-
>param('name');

$senderName = format_name($senderName);

#
# Crea un objeto MIME::Lite::HTML y lo inicializa
# con referrer
my $mailHTML = new MIME::Lite::HTML
                From => $senderName . '<' .
                               $cfg->val($pageName, 'from') . '>',
                To    => $to,
                Subject => $cfg->val($pageName, 'subject');

#
# Analiza la URL referrer y crea un objeto de correo MIME
#
my $MIMEEmail = $mailHTML->parse($referrerURL);

#
# envía el correo MIME
#

```

```

$MIMEEmail->send;

#
# Comprueba la cadena de error para buscar errores
#
my @errors = $mailHTML->errstr;

#
# Si no encuentra error, le dice al usuario que esa página fue
# enviada por su amigo; en caso contrario, muestra un mensaje de
error
#
if ($#errors > -1 ) {
    print abort('Error(s) found: ' . join('\\n', @errors));
} else {
    print abort("Page sent to $to.\\nThank you.");
}

#
# Finaliza
#
exit 0;

sub abort {
    # Imprime un mensaje javascript de abandono
    print sprintf("<script>alert('%s');history.go(-1);</
script>",shift);
    exit 0;
}

sub get_form_name {
    my $referrer = shift;

    # Si el HTTP tiene una cadena de consulta devuelve
    # únicamente la parte de la cadena que no es de consulta
    #
    if ($referrer =~ /\?/) {

        ($referrer, undef) = split(/\?/, $referrer);

    }

    return basename($referrer);
}

sub format_name {
    my @str = ();
    foreach my $part (split(/\s/,shift)) {

```

```

        push(@str, ucfirst(lc($part)));
    }

    return join(' ',@str);
}

sub check_email {
    my $email = shift;
    my $fieldName = shift;

    # Nota especial
    # Esta rutina de comprobación del e-mail debería devolver
    # false (0) para las direcciones de e-mail raras compatibles
    # con RFC que no son habituales. Por ejemplo, bob&sandra@
    # domain.com fallará. Si quiere permitir este tipo de
    # e-mails "técticamente" válidos, debe modificar las
    # expresiones regulares utilizadas en esta subrutina.

    # dirección de e-mail
    $email =~ /(\S+)\@([\w\.-]+)/;
    $email = $1 . '@' . $2;

    # Almacena el valor intacto en el objeto de consulta
    $query->param($fieldName, $email);

    # Rompe el e-mail user@host en user y host
    my ($user, $host) = split(/@/, $email);

    # Devuelve false si la parte host no tiene
    # formato hostname.domain.tld o domain.tld
    #
    return ($host !~ /(^[a-z0-9\._]+\.[a-z]{2,3})$/i) ? 0 : 1;
}

```

A continuación tenemos lo que está ocurriendo en este script:

- El script crea un objeto CGI llamado \$query y carga el archivo de configuración mime-mail.ini en otro objeto llamado \$cfg. Entonces almacena la información de la referencia HTTP en la variable llamada \$referrerURL y utiliza la subrutina get\_form\_name() para determinar el nombre de la página que llamó al script. El nombre de la página se almacena en la variable \$pageName.
- La cabecera de contenido por defecto (text/html) se imprime utilizando el método \$query->header. Entonces comprueba si hay una sección en la configuración para dicha página. Utiliza el método \$cfg->val(\$pageName, 'from') para localizar un parámetro llamado 'from' en el archivo de configuración mime-mail.ini. Este parámetro debe asignarse en la sección apropiada para la página. A continuación tenemos el archivo de configuración mime-mail.ini.

```

[default]
subject = A Web page forwarded by a friend
from . = webmaster@domain.com

[mybooks.html]
subject = Check out Kabir's linux books!
From = webmaster@domain.com

```

Por ejemplo, si se llamó al script CGI desde una URL del tipo `http://www.domain.com/mybooks.html`, entonces el script asignará \$pageName a 'mybooks.html' y buscará el parámetro 'from' en la sección [mybooks.html] utilizando la llamada al método `$cfg->val($pageName, 'from')`. Si no está el parámetro porque no se ha asignado o porque la sección no está, se fija el \$pageName en 'default', que le permite utilizar los parámetros de la sección [default]. De ese modo, tener una sección [default] es una buena idea si no quiere definir una sección para cada página en la que mostrar el formulario para enviar una página por correo electrónico.

- A continuación, el script determina si el usuario ha introducido una dirección de correo electrónico completa, y si es así, es decir, si la dirección de correo electrónico se encuentra en el formato `user@host`. Si no hay dirección de correo electrónico o no es válida, se muestra un mensaje de error y el script termina utilizando la subrutina `abort()`. Si la dirección de correo electrónico es válida, comprueba si el usuario ha introducido su nombre en el campo name del formulario Web. Si no hay nombre, abandona.
- Si se aceptan la dirección de correo electrónico y el nombre, el script da formato al nombre de usuario utilizando la subrutina `format_name()` y crea el objeto MIME::Lite::HTML. Este objeto se inicializa con información From (de), To (para) y Subject (asunto). La dirección From se compone del nombre de usuario y de la dirección de correo electrónico leídos desde el archivo de configuración. Por ejemplo, si un usuario llamado John Olson rellena el formulario para enviar una página por correo electrónico en la página `http://www.domain.com/mybooks.html`, el script creará la cabecera From como `John Olson <webmaster@domain.com>`. La dirección de correo electrónico del usuario se utiliza en el nombre para decirle al amigo o al compañero quién le está enviando el correo. También hemos recogido la dirección de correo electrónico del usuario, de modo que podemos utilizarlo para determinar la cabecera From. Pero recopilar la dirección de correo electrónico del usuario y del amigo o compañero en dos pequeñas cajas de texto podría ser muy confuso sin las etiquetas adecuadas, las cuales podrían variar el estado real de su página; por eso, evitamos este tipo de esquemas.
- El objeto \$mailHTML se utiliza entonces para analizar la URL de referencia. En este momento, el script CGI realmente actúa como un cliente

Web embebido y recupera la página Web de referencia y crea el contenido MIME necesario para enviar el correo electrónico. El método \$mailHTML->send se utiliza para enviar el correo electrónico. El método \$mailHTML->Estr. se utiliza para detectar errores y almacenarlos en @errors. Si no hay errores, el script muestra una página enviando el mensaje y termina.; si hay errores, muestra un mensaje de error y termina.

## Permitir soporte de depuración de errores CGI en Apache

Para ayudar a los desarrolladores de CGI, Apache tiene registros para las salidas CGI. Para cada error en el programa CGI, el archivo de registro contiene unas cuantas líneas de entradas de registro. Las primeras dos líneas contienen el momento de la solicitud, la URI solicitada, el estado HTTP, el nombre del programa CGI, y similares. Si no se puede ejecutar el programa CGI, hay dos líneas adicionales que contienen información sobre el error. De forma alternativa, si el error es el resultado de una información incorrecta sobre la cabecera, la información se registra como: todas las cabeceras de solicitudes HTTP, todas las cabeceras resultantes del programa CGI, y STDOUT y STDIN del programa CGI. Si el script falla en la salida, no se incluye el STDOUT.

Para registrar salidas CGI en Apache, utilice las directivas descritas en las siguientes secciones del módulo mod\_cgi, que forma parte de la distribución estándar. Con estas directivas puede determinar el registro de programas CGI que está desarrollando o intentando instalar en su sistema.

## ScriptLog

La directiva ScriptLog asigna los nombres de los archivos de registro para los errores del programa CGI. Si el nombre del archivo de registro es relativo (es decir, no comienza con una barra inclinada /), se toma como relativo con respecto al directorio raíz del servidor asignado por la directiva ServerRoot.

**Sintaxis:** ScriptLog filename

**Contexto:** configuración de recursos

**ADVERTENCIA:** Cuando utilice esta directiva, asegúrese de que el nombre especificado en la directiva UserDir coincide con el nombre del directorio de registros. Utilizar esta directiva en la configuración de recursos es una idea ya que se pierde información importante. La mejor manera de usarla es dentro de un script CGI, y devolver el resultado a través de la salida de STDOUT.

## **ScriptLogLength**

La directiva `ScriptLogLength` limita el tamaño del archivo de registro especificado en la directiva `ScriptLog`. El archivo de registro puede registrar una gran cantidad de información para cada error CGI, y por lo tanto, puede crecer rápidamente. Utilizando esta directiva, puede limitar el tamaño de registro de modo que cuando el archivo alcanza el máximo tamaño, no se registra más información.

**Sintaxis:** `ScriptLogLength size`

**Predefinido:** `ScriptLogLength 10385760`

**Contexto:** configuración de recursos

## **ScriptLogBuffer**

La directiva `ScriptLogBuffer` limita el tamaño de los datos POST o PUT que se registran.

**Sintaxis:** `ScriptLogBuffer size`

**Predefinido:** `ScriptLogBuffer size 1024`

**Contexto:** configuración de recursos

## **Depurar errores en sus scripts basados en Perl**

Si utiliza scripts CGI basados en Perl, tal y como se ha discutido en este capítulo, tiene muchas más ayudas en la solución de problemas que si simplemente utiliza lo que ofrece Apache como registros CGI. Puede depurar errores en scripts CGI basados en Perl desde la línea de comandos utilizando el famoso módulo `CGI.pm`.

O puede escribir mensajes de depuración de errores en el archivo de registro estándar de errores (`STDERR`), que redirige automáticamente al registro de errores de Apache. Veremos estas técnicas en las secciones siguientes.

## **Depuración de errores desde la línea de comandos**

Si utiliza el módulo `CGI`, tal y como hice yo en todos los scripts que hemos visto en este capítulo, está de suerte. El módulo `CGI` le permite solucionar problemas en su script CGI desde la línea de comandos, lo que es realmente útil para

depurar un script. Vamos a ver un script CGI llamado badcalc.pl, que se muestra en el listado 12.7.

**Listado 12.7. badcalc.pl**

```
#!/usr/bin/perl -w

use CGI;

my $query = new CGI;

my $num1 = $query->param('num1');
my $num2 = $query->param('num2');

my $sum = $num1 + num2;

#imprime $query->header;

print "$num1 + $num2 = $sum";

exit 0;
```

Cuando se accede a este script mediante una URL como `http://www.domain.com/cgi-bin/notready.pl`, devuelve un mensaje interno de error al servidor y registra un mensaje de error en el archivo de registro de errores del servidor.

Vamos a ver por qué funciona este pequeño script. A continuación tenemos una sesión típica de depuración de errores.

1. Permite la depuración de errores desde la línea de comandos para el módulo CGI cambiando la utilización de la línea `use CGI`:

```
use CGI qw(-debug);
```

Esto permite la depuración de errores en la línea de comandos para el módulo.

2. Como `root` determine el usuario Apache (es decir, el usuario que asignó a la directiva `User`) y ejecute el script desde la línea de comandos, verá este mensaje:

```
(offline mode: enter name=value pairs on standard input)
```

y el script esperará a que introduzca algún dato.

3. En la línea de comandos, introduzca pares clave=valor en cada línea para simular la entrada desde la Web. Por ejemplo, para alimentar el script anterior, este podría ser un ejemplo de una sesión en la línea de comandos:

```
(offline mode: enter name=value pairs on standard input)
num1=100
num2=200
```

Este código fija el campo de la entrada num1 en 100 y el campo de la entrada num2 en 200. Cada campo tiene un valor determinado en su propia línea.

4. Cuando esté realizando todas las entradas, presione Ctrl+D para determinar la parte de la entrada de la depuración y ver lo que hace el script. La sesión completa de depuración para la entrada anterior se muestra a continuación:

```
(offline mode: enter name=value pairs on standard input)
num1=100
num2=200
[control+d]
100 + 200 = 100
```

Como puede ver, el script añadió dos números e imprimió los datos tal y como se esperaba. Por lo tanto, ¿por qué fracasa este script cuando se ejecuta desde la Web? Bien, ¿ha visto alguna cabecera Content-Type antes del resultado? No. Si observa el script observará esa \$query->header; línea comentada. Si elimina el comentario y vuelve a ejecutar el script en el modo de línea de comandos, verá lo siguiente:

```
(offline mode: enter name=value pairs on standard input)
num1=100
num2=200
Content-Type: text/html; charset=ISO-8859-1

100 + 200 = 100
```

## Depuración utilizando la impresión de registros y de depuración

Este tipo de depuración de errores en la línea de comandos es muy útil para scripts pequeños y poco complicados, pero si tiene un script más grande, como el formwizard.pl, la depuración de errores desde la línea de comandos es muy engorrosa. En estos casos tenemos que utilizar una combinación de impresión de registro y de depuración. A continuación tenemos un script ejemplo, llamado calc.pl, que utiliza la impresión de registro y de depuración:

```
!/usr/bin/perl -w

use CGI qw(-debug);

use constant DEBUG => 1;

my $query = new CGI;

my $num1 = $query->param('num1');
```

```

my $num2 = $query->param('num2');

print $query->header;

if ($num1 == $num2) {

    # algo útil
    DEBUG and print STDERR "num1 and num2 are same.\n";

} elsif ($num1 > $num2) {

    # algo útil
    DEBUG and print STDERR "num1 is greater than num2.\n";

} elsif ($num1 < $num2) {

    # algo útil
    DEBUG and print STDERR "num1 is less than num2\n";

}

print $query->start_html('Calculator');
print $query->h1("Calculator");
print $query->p("Number 1: $num1");
print $query->p("Number 2: $num2");
print $query->end_html;

exit 0;

```

Cuando llamamos a este script desde una URL como `http://www.domain.com/cgi-bin/calc.pl?num1=100&num2=300`, imprime la información en el registro de error estándar de ese sitio. Para la URL mencionada, la entrada en el registro de error será parecida a esta:

```
[Tue Mar 20 20:04:26 2001] [error] [client 207.183.233.19] num1 is less than num2
```

La siguiente sentencia imprime este mensaje de error:

```
DEBUG and print STDERR "num1 is less than num2\n";
```

La parte interesante de esta línea es que utiliza una constante llamada `DEBUG`, que está asignada al comienzo del script con esta línea:

```
use constant DEBUG => 1;
```

La lógica de la sentencia `DEBUG and print` es la siguiente:

- Cuando `DEBUG` tiene el valor 1 o un número que no sea cero, es equivalente al valor '`true`' obtenido cuando se utiliza `DEBUG` en una operación lógica.
- La función integrada de impresión siempre devuelve un valor distinto de cero cuando la impresión tiene éxito.

- Por eso, cuando Perl evalúa DEBUG and print, ejecuta la sentencia de impresión.
- Cuando DEBUG tiene el valor 0, la sentencia DEBUG and print no se ejecuta.

Esto le permite insertar sentencias print que pueden formar parte de su código pero que pueden desactivarse cuando está realizando la depuración de errores. Observe que la sentencia print escribe en STDERR, que siempre escribe los datos en los registros de error para el sitio Web.

Para desactivar estas sentencias, simplemente fije la constante DEBUG en 0. Ahora, alguien podría decir que deberíamos eliminar totalmente estas sentencias de su script cuando esté preparado para manejar el script para producción. El razonamiento que encontramos bajo este tipo de argumento es que Perl sigue evaluando estas sentencias DEBUG incluso aunque no imprima nada, haciendo el script más lento. La verdad es que en una solución CGI, la diferencia de velocidad no importa porque los scripts CGI ya tienen una sobrecarga superior de la que tiene un mod\_perl u otra solución persistente. Pero si no quiere preocuparse, elimine las sentencias DEBUG antes de enviar el script a producción.

## Depurar con CGI::Debug

A continuación vamos a ver otro tipo de solución para la depuración de errores. Puede obtener una gran cantidad de ayuda en la depuración de errores de sus programas CGI utilizando el módulo CGI::Debug. Simplemente añada este módulo justo detrás de la sentencia use CGI; en su script, y podrá cazar todos los errores posibles. Por ejemplo:

```
!/usr/bin/perl -w

use CGI;
use CGI::Debug;

my $query = new CGI;

my $num1 = $query->param('num1');
my $num2 = $query->param('num2');

#imprime $query->header;

print $query->start_html('Calculator');
print $query->h1("Calculator");
print $query->p("Number 1: $num1");
print $query->p("Number 2: $num2");
print $query->end_html;

exit 0;
```

He comentado la línea \$query->header intencionadamente, la cual normalmente generaría un mensaje de error interno del servidor en el navegador Web. Por eso he añadido la sentencia use CGI::Debug; en este script, el script mostrará lo siguiente cuando se acceda a él como http://www.domain.com/c/s.dll/cgidebug.pl?num1=1&num2=200:

```
/cgi-bin/cgidebug.pl
```

```
Malformed header!
```

```
-- Program output below -----
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
    "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-US"><head><title>Calculator</title>
</head><body><h1>Calculator</h1><p>Number 1: 1</p><p>Number 2: 200</p></body></html>
-----
```

```
This program finished in 0.078 seconds.
```

```
Parameters
-----
num1 = 1[1]
num2 = 3[200]
```

```
Cookies
-----
```

```
Environment
-----
DOCUMENT_ROOT      = 15[/home/kabir/www]
GATEWAY_INTERFACE = 7[CGI/1.1]
HTTP_ACCEPT        = 133[image/gif, image/x-xbitmap, image/jpeg,
image/pjpeg, applica]...
HTTP_ACCEPT_ENCODING = 13[gzip, deflate]
HTTP_ACCEPT_LANGUAGE = 5[en-us]
HTTP_CONNECTION    = 10[Keep-Alive]
HTTP_HOST          = 14[rhat.nitec.com]
HTTP_USER_AGENT    = 50[Mozilla/4.0 (compatible; MSIE 5.5;
Windows NT 5.0)]
PATH               = 60[/usr/bin:/bin:/usr/sbin:/sbin:/usr/X11R6/bin:/home/kabir/bin]
QUERY_STRING        = 15[num1=1&num2=200]
REMOTE_ADDR        = 14[207.183.233.19]
REMOTE_PORT         = 4[2841]
REQUEST_METHOD     = 3[GET]
```

```
REQUEST_URI          = 36[/cgi-bin/cgidebug.pl?num1=1&num2=200]
SCRIPT_FILENAME     = 37[/home/kabir/www/asb2/ch12/cgidebug.pl]
SCRIPT_NAME         = 20[/cgi-bin/cgidebug.pl]
SERVER_ADDR         = 14[207.183.233.20]
SERVER_ADMIN        = 16[you@your.address]
SERVER_NAME         = 14[rhat.nitec.com]
SERVER_PORT          = 2[80]
SERVER_PROTOCOL      = 8[HTTP/1.1]
SERVER_SIGNATURE     = 66[<ADDRESS>Apache/2.0.14 Server at
rhat.nitec.com Port 80</ADD>...]
SERVER_SOFTWARE       = 20[Apache/2.0.14 (Unix)]
```

<EOF>

Como puede ver, hay una gran cantidad de información que le ayudará a solucionar problemas y arreglar el script rápidamente. Por ejemplo, una línea en el programa anterior indica que la cabecera está mal formada.





# 13 Server Side Includes (SSI)

---

## En este capítulo

1. Intentamos comprender los Server Side Includes.
2. Establecemos Apache para Server Side Includes.
3. Aplicamos Server Side Includes en páginas Web.

En el capítulo 12, hemos visto que el contenido dinámico se puede crear utilizando programas CGI; sin embargo, hay tareas que no van a requerir programas CGI a gran escala pero que siguen necesitando alguna intervención dinámica.

Por ejemplo, imagine que quiere añadir un mensaje copyright estándar en todas sus páginas HTML; ¿cómo implementaría esta posibilidad? Tiene dos soluciones:

- Añadir el contenido del mensaje copyright en cada página HTML.
- Escribir un programa CGI que añada el mensaje en cada página HTML.

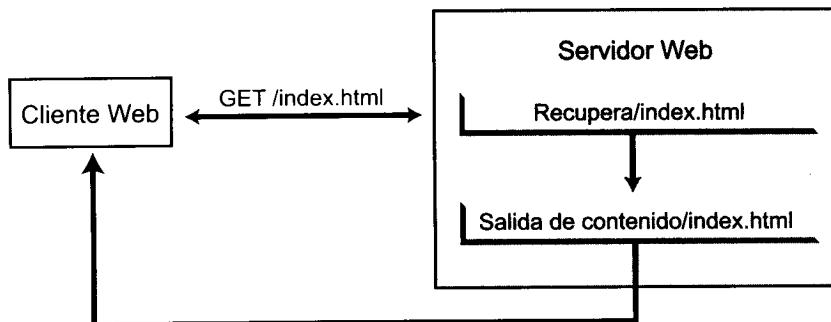
Ninguna de estas dos opciones resulta elegante. La primera opción requiere que cada vez que quiera cambiar el mensaje copyright, tenga que actualizar manualmente todos sus archivos. La segunda opción requiere que tenga algún modo

de ejecutar su programa CGI antes de enviar cada página al navegador Web. Esto también significa que el enlace de cada página tiene que llamar a este programa CGI de modo que puede adjuntar el mensaje a la siguiente página. Este tipo de situaciones demanda una solución sencilla. Server Side Include (SSI), el asunto de este capítulo, es esta solución sencilla.

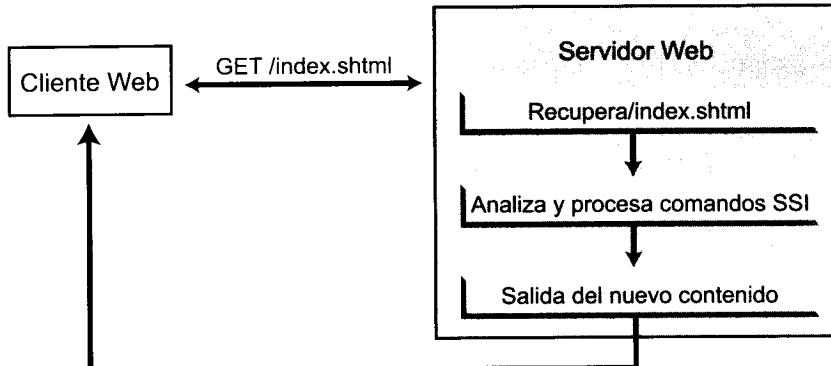
## Server Side Include

Normalmente, una página SSI es una página HTML con comandos embebidos para el servidor Web de Apache. El servidor Web normalmente no analiza páginas HTML antes de enviarlas al navegador Web (o a cualquier otro cliente Web). Sin embargo, antes del envío, el servidor Web siempre analiza una página HTML que tenga SSI, y si encuentra en la página un comando especial de SSI, lo ejecuta. La figura 13.1 muestra un ciclo de envío simplificado de una página HTML y de una página HTML con SSI (SHTML) desde un servidor Web.

Ciclo simplificado de envíos HTML



Ciclo simplificado de envíos SHTML



**Figura 13.1.** Diagramas simplificados de ciclos de envíos de una página HTML y una SHTML

Como puede ver, la versión SSI de la página HTML es analizada primero por los comandos SSI. Estos comandos se ejecutan, y la nueva salida se envía la navegador Web (es decir, al cliente Web).

Apache implementa SSI como un filtro `INCLUDES`. Antes de que pueda configurar Apache para SSI, tiene que comprobar su ejecutable actual de Apache (`httpd`) para asegurarse de que el módulo `mod_include` está incluido. Le mostraré cómo hacerlo en la siguiente sección.

## Configurar Apache para SSI

Antes de utilizar SSI en Apache, tiene que estar seguro de que está permitido el soporte SSI. Para determinar si tiene construido el `mod_include` en su binario Apache actual, ejecute el comando `httpd -l | grep include` desde el directorio `/usr/local/apache/bin` o desde donde tenga instalados los binarios de Apache. Esto le permite ver la lista de todos los módulos utilizados en la construcción de su ejecutable de Apache. Por defecto, debería tener este módulo compilado; si no es así, tiene que configurar la fuente de Apache utilizando la opción `--enable INCLUDES` y volviendo a compilar y a instalar Apache.

Aunque el módulo `mod_include` está compilado por defecto en la distribución estándar de Apache, el análisis de las páginas HTML no está activado por defecto. Puede activar SSI para un directorio completo o para un solo tipo de archivo, tal y como se verá en la sección siguiente.

## Activar SSI para un directorio completo

Para activar SSI para un directorio llamado `/www/mysite/htdocs/parsed`, añada la siguiente configuración a `httpd.conf`:

```
<Directory "/www/mysite/htdocs/parsed">
    Options +Includes
    SetOutputFilter INCLUDES
</Directory>
```

Aquí, la directiva `Options` está asignada a `+Includes`, que permite el análisis SSI en este directorio. La directiva `SetOutputFilter` le dice a Apache que analice todas las páginas desde este directorio para comandos SSI. Esto significa que los archivos con cualquier extensión en este directorio serán analizados por el servidor.

Por ejemplo, imagine que tiene la siguiente configuración de host virtual:

```
<VirtualHost 192.168.1.100>
    ServerName vh1.domain.com
    DocumentRoot "/www/mysite/htdocs"
```

```
ScriptAlias /cgi-bin/ "/www/mysite/htdocs/cgi-bin/"

<Directory "/www/mysite/htdocs/parsed">
    Options +Includes
    SetOutputFilter INCLUDES
</Directory>

</VirtualHost>
```

Ahora, si el directorio `/www/mysite/htdocs/parsed` tiene algún archivo `any.txt`, `any.html` o `any.shtml` en él, se analizarán estas solicitudes URL:

```
http://vh1.domain.com/parsed/any.txt
http://vh1.domain.com/parsed/any.html
http://vh1.domain.com/parsed/any.shtml
```

En la mayoría de los casos, no es necesario analizar un archivo de texto (`.txt`) o un archivo html (`.html`) porque estos archivos no se utilizan normalmente para los comandos SSI. Por eso, la configuración anterior hará que Apache funcione a no ser que quiera analizar todos los tipos de archivos. A continuación vamos a ver cómo puede limitar el análisis a un determinado tipo de archivos.

## Activar SSI para un tipo específico de archivo

Para limitar el alcance del análisis SSI en un directorio, utilice la directiva `AddType` para determinar la cabecera `Content-Type` que necesitamos para el tipo de archivo que admite SSI y entonces meta el filtro `INCLUDES` en un contenedor `FilesMatch`. Por ejemplo:

```
Options +Include
AddType text/html .shtml

<FilesMatch "\.shtml[.]$">
    SetOutputFilter INCLUDES
</FilesMatch>
```

Aquí, la directiva `Options` está asignada a `+Includes`, que permite el análisis SSI. La directiva `AddType` se utiliza para asignar la cabecera `Content-Type` para un tipo de archivo llamado `.shtml` a `text/html`. Entonces la directiva `SetOutputFilter` se asigna a `INCLUDES` para los archivos `.shtml` utilizando la directiva `FilesMatch` y una expresión regular `"\.\shtml[.]$"`.

A continuación vamos a volver al host virtual del ejemplo de la sección anterior. Esta vez vamos a añadir el contenedor `FilesMatch`:

```
<VirtualHost 192.168.1.100>

    ServerName vh1.domain.com
```

```

DocumentRoot "/www/mysite/htdocs"
ScriptAlias /cgi-bin/ "/www/mysite/htdocs/cgi-bin/"

<Directory "/www/mysite/htdocs/parsed">
    Options +Includes

    AddType text/html .shtml

    <FilesMatch "\.shtml[.]$">
        SetOutputFilter INCLUDES
    </FilesMatch>

</Directory>

</VirtualHost>

```

Ahora, si hay archivos `any.txt`, `any.html` o `any.shtml` en el subdirectorio analizado, la siguiente URL le dirá a Apache que analice sólo la salida del archivo `.shtml`.

`http://vh1.domain.com/parsed/any.shtml`

El servidor no analizará las otras dos URL, `http://vh1.domain.com/parsed/any.txt` y `http://vh1.domain.com/parsed/any.html`, para comandos SSI.

Esta es la configuración preferida en la mayoría de los sitios porque podemos limitar el análisis a un tipo determinado de archivos con propósitos de rendimiento y de organización del sitio.

**ADVERTENCIA:** Si tiene pensado desactivar la ejecución de programas externos mediante comandos SSI, puede utilizar la opción `IncludesNOEXEC` con la directiva `Options`. Esto desactiva la ejecución de programas externos. Sin embargo, también desactiva la carga de archivos externos mediante el comando `SSI Include`.

## Utilizar XBitHack para archivos .htm o .html

Tal y como se mencionó antes, activar el análisis SSI para un directorio completo, disminuye el rendimiento del servidor. Podría intentar evitar la utilización de extensiones `.html` o `.htm` para SSI; si tiene que utilizarlos, entonces utilice la directiva `XbitHack` que se encuentra en el módulo `mod_include`. La directiva `XBitHack` controla el análisis de archivos asociados con el tipo MIME `text/html`:

**Sintaxis:** `XBitHack On | Off | Full`

**Predefinido:** `XBitHack Off`

**Contexto:** configuración del servidor, host virtual, directorio, archivo de control de acceso en el ámbito de directorio (.htaccess)

### **Invalidar:** Options

Normalmente, únicamente los archivos .html y .htm están asociados con text/html. El valor por defecto off le dice al servidor que no analice estos archivos. Cuando está fijado con el valor on, cualquier archivo HTML que tiene permiso de ejecución para el dueño del archivo, se considera un archivo SSI y se analiza. Cuando la directiva tiene el valor full, hace que el servidor compruebe los bits ejecutables del dueño y del grupo de las asignaciones de permisos para el archivo. Si está asignado el bit ejecutable del grupo, entonces Apache asigna la fecha de la última modificación del archivo devuelto para que sea el momento de la última modificación del archivo. Si no está asignado, entonces no se envía fecha de última modificación. Asignar este bit permite a los clientes y a los proxies cachear el resultado de la solicitud. Utilizar el valor full no está recomendado para páginas SSI que producen una salida distinta cuando son analizadas y procesadas.

**NOTA:** Seguirá utilizando Options +Includes cuando utilice la directiva XBitHack para permitir soporte SSI.

Si utiliza el archivo de control de acceso en el ámbito de directorios (.htaccess) para permitir soporte SSI, asegúrese de que esa directiva AllowOverride para el sitio, pertenece al directorio que permite esa operación. La directiva AllowOverride para estos sitios debe permitir la invalidación de la opción Includes. Por ejemplo, si la directiva AllowOverride tiene asignado el valor None para un sitio, no tendrá lugar el análisis SSI.

**NOTA:** Si no quiere utilizar el signo + en la línea Options del ejemplo anterior, todas las opciones estarán desactivadas excepto Includes.

Ahora que sabe cómo permitir soporte SSI en Apache, la siguiente sección discute los comandos SSI en detalle.

## **Utilizar comandos SSI**

Los comandos SSI están embebidos en páginas HTML en forma de comentarios. La estructura básica de un comando es la siguiente:

```
<!--#command argument1=value argument2=value argument3=value -->
```

El valor suele estar encerrado entre comillas; muchos comandos permiten un solo par atributo-valor. Observe que la terminación de comentario --> suele ir

precedida por un espacio en blanco para asegurar que no se considera parte del comando SSI.

Las siguientes secciones examinan todos los comandos SSI disponibles.

## config

El comando `config` le permite configurar el mensaje de error que aparece, así como el formato que se utiliza para mostrar la información de tiempo y tamaño. Esto se lleva a cabo con las siguientes líneas de código:

```
config errmsg="error message"
config sizefmt=["bytes" | "abbrev"]
config timefmt=format string
```

`config errmsg="error message"` le muestra cómo crear un mensaje de error personalizado, que se muestra cuando tiene lugar un error en el análisis. Por ejemplo, el listado 13.1 muestra un archivo llamado `config_errmsg.shtml`.

**Listado 13.1.** config\_errmsg.shtml

```
<HTML>
<BODY>
    <TITLE> Apache Server 2 - Chapter 13 </TITLE>
</HEAD>

<BODY BGCOLOR="white">

<FONT SIZE=+1 FACE="Arial"> Simple SSI Example #1</FONT>
<HR SIZE=1>

<P> Example of the SSI <STRONG>config errmsg</STRONG> command:
</P>
<P> Embedded commands: <BR><BR>

<CODE>
    &lt;!-#config errmsg="SSI error! Please notify the
webmaster." -&gt; <BR>
    &lt;!-#config badcommand="whatever" -&gt;
</CODE>

</P>

<P> Result: <BR>

<!--#config errmsg="SSI error! Please notify the Web master." -
->
<BR>
<!--#config badcommand="whatever" -->

</P>
```

```
</BODY>
</HTML>
```

En este archivo, hay dos comandos SSI:

```
<!--#config errmsg="SSI error! Please notify the webmaster." -->
y
<!--#config badcommand="whatever" -->
```

El primer comando es un comando config errmsg válido que asigna un mensaje de error a la cadena "SSI error! Please notify the Web master". El segundo comando es un comando SSI no válido, que he introducido intencionadamente en este archivo, para que pudiera ver qué ocurre cuando Apache lo analiza. La figura 13.2 muestra lo que devuelve el navegador cuando el servidor analiza esta página.

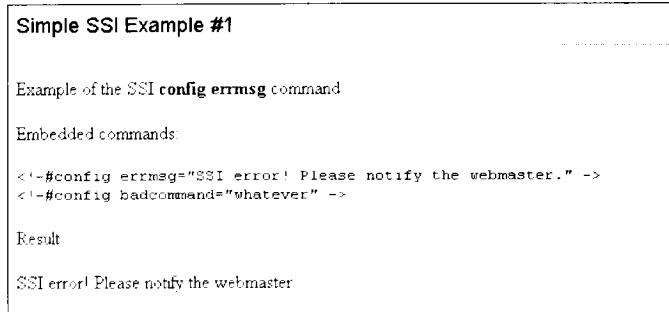


Figura 13.2. Ejemplo del comando config errmsg

Como puede ver en la figura, el segundo comando da lugar a un error, y el mensaje de error se muestra como resultado. El mensaje aparece donde se encuentra el comando.

**NOTA:** Puede introducir etiquetas HTML o incluso insertar un script del lado del cliente en la cadena del mensaje de error. Por ejemplo, el siguiente código, muestra una ventana de alerta de JavaScript con un mensaje de error:

```
<!--#config errmsg=<SCRIPT LANGUAGE=JavaScript> alert('An error
occurred. \n Please report to webmaster@domain.com');</SCRIPT>-->
```

config sizefmt=[ "bytes" | "abbrev" ] le permite elegir el formato de salida para el tamaño del archivo. Los especificadores de formato aceptables son "bytes" o "abbrev". Por ejemplo:

```
<!--#config sizefmt="bytes" -->
```

muestra el tamaño del archivo en bytes. Para mostrar los archivos en kilobytes o megabytes, utilice:

```
<!-#config sizefmt="abbrev" -->
```

config timefmt=format string le permite elegir el formato para el tiempo:

```
config timefmt=format string
```

Los valores utilizados normalmente para la cadena de formato pueden ser los identificadores mostrados en la tabla 13.1.

**Tabla 13.1.** Identificadores de formato para config timefmt

Identificador	Significado
%a	El nombre del día de la semana abreviado de acuerdo con el escenario actual.
%A	El nombre completo del día de la semana de acuerdo con el escenario actual.
%b	El nombre del mes abreviado de acuerdo con el escenario actual.
%B	El nombre completo del mes de acuerdo con el escenario actual.
%c	La mejor representación de la fecha y el tiempo para el escenario actual.
%d	El día del mes como un número decimal (del 01 al 31).
%H	La hora como un número decimal utilizando un reloj de 24 horas (de 00 a 23).
%I	La hora como un número decimal utilizando un reloj de 12 horas (de 01 a 12).
%j	El día del año como un número decimal (del 001 al 366).
%m	El mes como un número decimal (del 01 al 12).
%M	El minuto como un número decimal.
%p	Puede ser a.m. o p.m., de acuerdo con el valor de la hora o el escenario.
%S	Los segundos como un número decimal.
%w	El día de la semana como un decimal, el domingo (Sunday) es el 0.

Identificador	Significado
%x	La representación de la fecha preferida para el escenario actual sin la hora.
%X	La representación del momento preferido para el escenario actual sin la fecha.
%y	El año como un número decimal sin el siglo (del 00 al 99).
%Y	El año como un número decimal incluyendo el siglo.
%Z	El nombre de la zona horaria o abreviatura.
%%	Un carácter %.

Por ejemplo, el siguiente código asigna el formato de tiempo del tipo Sat Mar 17 00:31:58 2001:

```
<!--#config timefmt="%c" -->
```

Y el siguiente, asigna el formato de tiempo del tipo 03/17/2001:

```
<!--#config timefmt="%m/%d/%Y" -->
```

## echo

El comando echo imprime una de las variables `Include` (que se definen más tarde) o alguna de las variables de entorno CGI. La sintaxis es:

```
echo var="variable_name"
```

Si el valor de esta variable no está disponible, no imprime nada. Cualquier fecha impresa es objeto del `timefmt` configurado. Por ejemplo:

```
<!--#config timefmt="%m/%d/%Y" -->
<!--#echo var="DATE_LOCAL" -->
```

que imprime una fecha del tipo 03/17/2001, en conformidad con la cadena `timefmt` especificada.

## exec

El comando exec le permite ejecutar un programa externo. El programa externo puede ser un programa CGI o cualquier otro tipo de programa de un ejecutable, como los scripts shell o los archivos binarios de datos. La sintaxis de los programas CGI es:

```
exec cgi="path_to_cgi_program"
```

La sintaxis para otros programas es:

```
exec cmd="path_to_other_programs"
```

**NOTA: Si utiliza el valor `IncludesNOEXEC` para la directiva `Options`, este comando está desactivado.**

Vamos a ver cómo utilizar cada una de estas opciones.

## cgi

El valor `cgi` determina una ruta URL relativa (%-encoded) para el script CGI. Si la ruta no empieza con una barra inclinada (/), se toma como relativa para el documento actual. El documento al que nos referimos con esta ruta se invoca como un script CGI, incluso si el servidor no lo reconociese normalmente como tal. Sin embargo, el directorio que contiene el script debe permitir scripts CGI (con `ScriptAlias` o la opción `ExecCGI Option`).

El script CGI ofrece el `PATH_INFO` y la cadena de consulta (`QUERY_STRING`) de la solicitud original del cliente; esto se puede especificar en la ruta de la URL. Las variables `Include` están disponibles para el script, además de para el entorno CGI estándar.

El listado 13.2 muestra un sencillo script CGI llamado `colors.pl`, que muestra una lista de colores habituales en una tabla HTML.

**Listado 13.2. colors.pl**

```
#!/usr/bin/perl -w

use strict;

my @COLOR_LIST = qw(red blue brown yellow green gray white
black);

print "Content-type: text/html\n\n";

print '<table border=1 cellpadding=3 cellspacing=0>';

foreach my $color (sort @COLOR_LIST) {

    print <<TABLE_ROW;
    <tr><td>$color</td>
        <td bgcolor="$color"> &ampnbsp &ampnbsp &ampnbsp &ampnbsp </td>
    </tr>

TABLE_ROW
}
```

```
print '</table>';

exit 0;
```

Ahora observe que este script se está llamando desde el archivo exec\_cgi1.shtml, que se muestra en el listado 13.3.

#### Listado 13.3. exec\_cgi1.shtml

```
<HTML>
<HEAD> <TITLE> Apache Server 2 - Chapter 13 </TITLE></HEAD>

<BODY BGCOLOR="white">
<FONT SIZE=+1 FACE="Arial">SSI Example #2</FONT>
<HR SIZE=1>

<P> Example of the SSI <STRONG>exec cgi</STRONG> command: </P>
<P> Embedded commands: <BR><BR>

<CODE> &lt;!--#exec cgi="/cgi-bin/colors.pl" --> <BR> </CODE>
</P>
<P> Result: <BR> <!--#exec cgi="/cgi-bin/colors.pl" --> </P>

</BODY>
</HTML>
```

Utilizando el comando <!--#exec cgi="/cgi-bin/colors.pl" -->, exec\_cgi1.shtml produce la salida que se muestra en la figura 13.3.

La belleza de embeber un script CGI utilizando una llamada SSI como la anterior radica en que la página se ha armado utilizando tanto datos estáticos y dinámicos (es decir, contenido del script CGI).

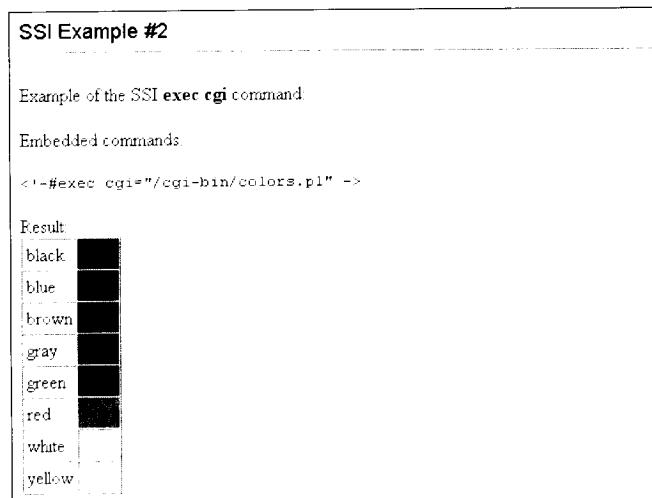


Figura 13.3. Salida del archivo exec\_cgi1.shtml

Observe que si el script CGI devuelve una cabecera Location en lugar de una salida, la cabecera se traduce a un anchor (marcador que se inserta en una página Web) HTML. Por ejemplo, el listado 13.4 muestra un script CGI escrito en Perl llamado `relocate.pl` que imprime una cabecera Location: como salida.

#### Listado 13.4. relocate.pl

```
#!/usr/bin/perl -w

print 'Location: http://apache.nitec.com' . "\n\n";

exit 0;
```

Cuando un navegador Web solicita el archivo `exec_cgi2.shtml`, mostrado en el listado 13.5, el servidor convierte la cabecera Location: en un anchor HTML en lugar de redirigir el navegador al sitio `http://apache.nitec.com`.

#### Listado 13.5. exec\_cgi2.shtml

```
<HTML>
<HEAD> <TITLE> Apache Server 2 - Chapter 13 </TITLE></HEAD>

<BODY BGCOLOR="white">
<FONT SIZE=+1 FACE="Arial">SSI Example #3</FONT>
<HR SIZE=1>

<P> Example of the SSI <STRONG>exec cgi</STRONG> command: </P>
<P> Embedded commands: <BR><BR>

<CODE> &lt;!--#exec cgi="/cgi-bin/relocate.pl" --> <BR> </
CODE>
</P>

<P> Result: <BR> <!--#exec cgi="/cgi-bin/relocate.pl" --> </P>

</BODY>
</HTML>
```

En el listado, la única llamada SSI en el archivo es:

```
<!--#exec cgi="/cgi-bin/relocate.pl" -->
```

La salida de esto es un anchor HTML, tal y como muestra la figura 13.4.

### cmd

Cuando llamamos a un programa distinto de un programa CGI, puede utilizar la versión cmd de la llamada exec. El servidor ejecuta la cadena dada utilizando el shell sh (/bin/sh) en la mayoría de los sistemas Unix. Las variables Include están disponibles para este comando. Por ejemplo, el listado 13.6 muestra un archivo llamado `exec_cmd.shtml`.

### SSI Example #3

Example of the SSI **exec cgi** command

Embedded commands:

```
<!--#exec cgi="/cgi-bin/relocate.pl" -->
```

Result:

<http://apache.nitec.com>

**Figura 13.4.** Salida del archivo exec\_cgi2.shtml

### Listado 13.6. exec\_cmd.shtml

```
<HTML>
<HEAD> <TITLE> Apache Server 2 - Chapter 13 </TITLE></HEAD>
<BODY BGCOLOR="white">

<FONT SIZE=+1 FACE="Arial"> Simple SSI Example #4</FONT>
<HR SIZE=1>

<P> Example of the SSI <STRONG>exec cmd</STRONG> command: </P>
<P> Embedded commands: <BR><BR>

<CODE>
&lt;!-#exec cmd="/bin/date +%m/%d/%y" -&gt; <BR>
&lt;!-#exec cmd="/bin/ls -l ." -&gt; <BR>
</CODE>

</P>
<P> Result: <BR>

<!--#exec cmd="/bin/date +%m/%d/%y" --> <BR>
<!--#exec cmd="/bin/ls -l ./*.html" --> <BR>

<PRE>
<!--#exec cmd="/bin/ls -l ./*.html" -->
</PRE>

</P>
</BODY>
</HTML>
```

Este archivo tiene dos llamadas cmd:

```
<!--#exec cmd="/bin/date +%m/%d/%y" -->
<!--#exec cmd="/bin/ls -l ./*.html" -->
```

La primera, llama a una utilidad Unix /bin/date con el argumento +%m/%d/%y; la segunda, llama a la utilidad Unix ls con ./\*.html como argu-

mento. La salida de este archivo se muestra en la figura 13.5. Observe que la salida ls está formateada utilizando el par <PRE> y </PRE>. Si quiere sacar algo que utilice esas líneas nuevas, podría utilizar etiquetas <PRE> para mantener la salida legible, tal y como muestra la figura 13.5.

```
Simple SSI Example #4

Example of the SSI exec cmd command

Embedded commands

<!--#exec cmd="/bin/date +\%m/\%d/\%y" -->
<!--#exec cmd="/bin/ls -l ./" -->

Result
03/17/01
```

Figura 13.5. Salida del archivo exec\_cmd.shtml

## fszie

Este comando imprime el tamaño del archivo especificado. La sintaxis que ha de utilizar para este comando depende de si la ruta del directorio es relativa o virtual:

```
fszie file="path"
fszie virtual="URL"
```

Cuando se utiliza la primera sintaxis, se supone que la ruta es relativa al directorio que contiene el documento SSI que se está analizando. No puede utilizar . . / en la ruta, ni puede utilizar rutas absolutas. No puede acceder a scripts CGI de ese modo. Puede, sin embargo, acceder a otro documento analizado. Por ejemplo:

```
<!--#fszie file="download.zip" -->
```

Si se utiliza la segunda sintaxis, se supone que la ruta virtual es una ruta URL (%-encoded). Si la ruta no comienza con una barra inclinada (/), entonces se toma como relativa al documento actual. Debe acceder a un archivo normal de este modo, pero no puede acceder a un script CGI así. De nuevo, sin embargo, puede acceder a otro documento analizado. Por ejemplo:

```
<!--#fszie virtual="/download/free_software.zip" -->
```

El formato de salida está sujeto a la especificación del formato sizefmt. Ver el comando config para obtener los detalles.

## flastmod

El comando flastmod imprime la última fecha de modificación del archivo especificado. De nuevo, hay dos opciones de sintaxis, dependiendo de la ruta del directorio:

```
flastmod file="path"  
flastmod virtual="URL"
```

La salida está sujeta a la especificación del formato de `timefmt`. Por ejemplo:

```
<!--#flastmod file="free_software.zip" -->  
<!--#flastmod virtual="/download/free_software.zip" -->
```

Si está confuso con respecto a la diferencia en la sintaxis, remítase al comando `fsize` como ejemplo. Para controlar cómo se imprime la modificación, ver el comando `config`.

## include

La directiva `include` inserta el texto de un documento en el documento SSI que se está procesando. La sintaxis depende de la ruta del directorio:

**Sintaxis 1:** `include file="path"`

**Sintaxis 2:** `include virtual="URL"`

Ver el comando `fsize` para observar las diferencias entre el modo de archivo y el virtual.

Cualquier archivo `included` es sujeto del control de acceso habitual. Si el directorio que contiene el archivo analizado, tiene asignado `Option IncludesNOEXEC`, e incluir el documento podría dar lugar a que se ejecutase el programa, entonces no se incluye. Esto previene la ejecución de scripts CGI. De lo contrario, los scripts CGI se invocan como siempre, utilizando la URL completa dada en el comando, incluyendo cualquier cadena de consulta. Por ejemplo:

```
<!--#include file="copyrights.html" -->
```

Incluye el archivo `copyrights.html` en el documento. Este comando es útil para añadir código HTML repetido en los archivos. Muchos sitios utilizan una barra de menú estándar en cada página; si esta barra de menú se coloca en un archivo HTML llamado `menu.html`, puede ser llamado desde todas las páginas SSI utilizando una llamada a un archivo `include`, parecida al del ejemplo anterior. En el futuro, cuando se necesiten realizar cambios en el menú, el administrador del sistema sólo tiene que actualizar la página `menu.html`. Esto ahorrará mucho tiempo si hay muchos archivos en el sitio.

Las inclusiones recurrentes se detectan y se genera un mensaje de error tras la primera pasada. Por ejemplo, si `a.shtml` tiene una llamada SSI del tipo:

```
<!--#include file="b.shtml" -->
```

y `b.shtml` tiene una llamada del tipo:

```
<!--#include file="a.shtml" -->
```

entonces Apache registra y muestra una indicación de error que advierte que se ha detectado una inclusión recurrente.

## printenv

El comando `printenv` imprime una lista de todas las variables y sus valores. La sintaxis es:

```
printenv
```

Por ejemplo:

```
<!--#printenv -->
```

imprime todas los `Include` y todas las variables de entorno CGI disponibles. Para facilitar la lectura de la salida, utilice el par de etiquetas `<PRE>`.

## set

El comando `set` asigna el valor de una variable definida por el usuario. La sintaxis es:

```
set var="variable name" value="value of the variable"
```

Por ejemplo:

```
<!--#set var="home" value="index.shtml" -->
```

# Variables SSI

El módulo SSI deja un conjunto de variables, además de las variables de entorno CGI (capítulo 12), disponibles para todos los archivos SSI. Estas variables se llaman `variables include`. Pueden ser utilizadas por comandos (`echo`, `if`, `elif`, y otros) y por cualquier programa invocado por un comando SSI. La tabla 13.2 muestra las variables `include`.

**Tabla 13.2.** Variables `Include`

Variable	Significado
<code>DATE_GMT</code>	La fecha actual en Greenwich.
<code>DATE_LOCAL</code>	La fecha actual en la zona horaria local.
<code>DOCUMENT_NAME</code>	El nombre del archivo SSI actual.
<code>DOCUMENT_URI</code>	La ruta URL (%-decoded) del documento.
<code>LAST_MODIFIED</code>	La última fecha de modificación del archivo actual. La fecha está sujeta al formato <code>timefmt</code> del comando <code>config</code> .

Las variables `include` y las variables CGI están programadas y disponibles para su utilización. Cualquiera de estas variables programadas, se puede utilizar como argumentos de otros comandos. La sintaxis para utilizar las variables definidas es:

```
<!--#command argument1="$variable1" argument2="$variable2" ... -->
```

Como puede ver, el nombre de la variable tiene el prefijo `$`. A continuación tiene otro ejemplo:

```
<!--#config errmsg="An error occurred in $DOCUMENT_NAME page." -->
```

Cuando utilizamos variables en un campo `var="variable"`, el signo `$` no es necesario. Por ejemplo:

```
<!--#echo var="DOCUMENT_NAME" -->
```

**NOTA: Si tiene que insertar el signo del dólar en el valor de una variable, puede insertar el signo del dólar utilizando la barra inversa. Por ejemplo:**

```
<!--#set var="password" value="\$cheese" -->  
<!--#echo var="password" -->
```

Esto imprime `\$cheese` como el valor de la variable "password".

Además, si necesita hacer referencia al nombre de una variable en el medio de una secuencia de caracteres que de otro modo podría considerarse un identificador válido, utilice un par de corchetes alrededor del nombre. Por ejemplo:

```
<!--#set var="uniqueid" value="${DATE_LOCAL}_${REMOTE_HOST}" -->
```

Esto fija `uniqueid` en algo parecido a Saturday, 17-Mar-2001 13:02:47 PST\_207.183.233.19, dependiendo de la asignación de `timefmt` y de la dirección IP del cliente Web.

## Control de flujo de los comandos

Al igual que en muchos lenguajes de programación, hay disponible un control del flujo del programa en el módulo SSI. Utilizando los comandos del control de flujo, puede crear diferentes salidas dependiendo de una serie de condiciones. La sentencia de control de flujo más sencilla (es decir, la condicional) es:

```
<!--#if expr="test_expression" -->  
<!--#endif -->
```

Aquí se evalúa "test\_expression", y si el resultado es verdadero, entonces todo el texto hasta el comando `endif` es incluido en la salida.

"test\_expression" puede ser un string, que es verdadero si la cadena no está vacía, o una expresión de comparación de valores de dos cadenas.

Los operadores de comparación permitidos son =, !=, <, >, <= o >=. Una forma genérica de estas sentencias SSI sería:

```
<!--#if expr="string1 operator string2" -->
<!--#endif -->
```

Observe que esa segunda cadena (string2) puede ser una expresión regular de la forma /regular expression patterns/ (/patrón de una expresión regular). Ver el apéndice B para obtener los detalles sobre las expresiones regulares.

Vamos a ver un ejemplo de un string:

```
<!--#if expr="foobar" -->
This test is always successful.
<!--#endif -->
```

Esta sintaxis siempre imprime This test is successful. porque la expresión es verdadera cuando test\_expression no es una cadena vacía. Si se cambia expr="foobar" a expr="" o a expr="''", entonces el texto dentro del bloque if-endif nunca formará parte de la salida.

Vamos a ver un ejemplo de una prueba de equivalencia de cadena:

```
<!--#set var="quicksearch" value="yes" -->
<!--#if expr="$quicksearch = yes" -->
    Quick search is requested.
<!--#endif -->
```

Aquí, la variable llamada quicksearch se fija con el valor yes, y luego se compara con yes. Como el valor asignado y el valor comparado es el mismo, la línea Quick search is requested será la salida.

Utilizando operadores de lógica como !, && y ||, puede crear test\_expressions (expresiones de prueba) más complicadas. Por ejemplo:

```
<!--#if expr="${REMOTE_ADDR} = /207\.183\.233/
&& ${DOCUMENT_NAME} = /timesheet/" -->
<!--#include virtual="/cgi-bin/timecard.pl">
<!--#endif -->
```

Aquí, la expresión de prueba se compone de dos pequeñas expresiones. La primera subexpresión, \${REMOTE\_ADDR} = /207\.183\.233/, se eva-

lúa para determinar si la variable definida por el servidor REMOTE\_ADDR corresponde a la dirección de red 207.183.233. Observe que la dirección está escrita utilizando la expresión regular /207\.183\.233/, en la que cada . (punto) se marca utilizando una barra invertida \. Esto era necesario para deshacer el significado concreto del punto en las expresiones regulares. Ver el apéndice C para obtener más detalles sobre las expresiones regulares.

La segunda subexpresión, \${DOCUMENT\_NAME} = /timesheet/, se evalúa para determinar si el archivo SSI actual que se está procesando, tiene un nombre que corresponda a la cadena timesheet. Y, finalmente, el && (el Y en lógica) exige que las dos subexpresiones sean ciertas para que la expresión sea cierta. Si la expresión final es cierta, entonces se ejecuta el script /cgi-bin/timecard.pl utilizando el comando virtual include.

Otras operaciones lógicas que puede realizar en la test\_expression son:

```
<!--#if expr="! test_expression" -->
This is printed only when the test_expression is false.
<!--#endif -->
```

y

```
<!--#if expr="test_expression1 || test_expression2" -->
This is printed when at least one of the test_expressions is
true.
<!--#endif -->
```

Los operadores = (igual) y != (distinto) tienen prioridad sobre los operadores && (y) y || (o). El operador ! (no) tiene la máxima prioridad. Puede utilizar un par de paréntesis para aumentar la prioridad. Por ejemplo:

```
<!--#if expr="($win = yes && $loss = false) != ($profit = yes)" -->
```

Aquí, (\$win = yes && \$loss = false) se evalúa antes que el operador !=.

Cualquier cosa que no se reconozca como variable o como un operador es tratado como una cadena. Las cadenas también pueden ir entre comillas simples: 'string'. Las cadenas o strings no pueden contener espacios en blanco (vacíos y tabulaciones) porque se utilizan para separar señales como son las variables. Si se encuentran varias cadenas en una fila, se concatenan utilizando vacíos.

Si necesita construcciones de control de flujo más complejas, puede utilizar:

```
<!--#if expr="test_condition1" -->
```

```
Do something specific to the first test condition.
```

```
<!--#elif expr="test_condition2" -->
```

```
Do something specific to the first second condition.
```

```

<!--#else -->
Do something as default.

<!--#endif -->
El elif le permite crear una condición else-if. Por ejemplo:
<!--#if expr="${HTTP_USER_AGENT} = /MSIE/" -->

<!--#set var="browser" value="MicrosoftIE" -->
<!--#include file="mypage.asp" -->

<!--#else -->

<!--#set var="browser" value="Others" -->
<!--#include file="mypage.html" -->

<!--#endif -->

```

Aquí, se comprueba la variable `HTTP_USER_AGENT` para determinar si contiene el string `MSIE` (un string utilizado por el navegador Microsoft Internet Explorer). Si contiene este string, entonces la variable del navegador se fija con el valor `MicrosoftIE`, y un archivo llamado `mypage.asp` se inserta en el documento. Sin embargo, si `HTTP_USER_AGENT` no contiene el string `MSIE`, supone que se trata de otro navegador (como Netscape Navigator, Lynx), y por lo tanto, la variable del navegador toma el valor `Others` y se inserta el archivo `mypage.html` en el documento. Utilizando la construcción `if-then-else`, este ejemplo asigna un valor distinto a la misma variable y carga distintos archivos.



# 14 Configurar Apache para FastCGI

---

## En este capítulo

1. Explicamos FastCGI.
2. Describimos la arquitectura básica de una aplicación FastCGI.
3. Compilamos e instalamos el módulo FastCGI para Apache.
4. Configuramos `httpd.conf` para que ejecute aplicaciones FastCGI.

Este capítulo discute FastCGI y resuelve los problemas de rendimiento inherentes a CGI, sin introducir la sobrecarga y la complejidad de los API propietarios. FastCGI es rápido, abierto y sostenible. Ofrece características como caching en memoria, conexiones persistentes y arquitectura distribuida. La migración desde CGI a FastCGI es además razonablemente simple.

## FastCGI

En general, una aplicación FastCGI actúa como una aplicación de servidor llamada demonio en el mundo Unix. A diferencia de un script CGI, una aplicación

FastCGI es persistente y sirve todas las solicitudes entrantes utilizando una única instancia, eliminando así, la sobrecarga inherente al hecho de iniciar un nuevo proceso para cada solicitud, como ocurre en la versión CGI. Por ejemplo, si tiene que generar un conjunto de páginas dinámicas utilizando contenido de una base de datos, un script CGI tiene que conectar con la base de datos cada vez que ejecuta una solicitud. Sin embargo, una aplicación FastCGI puede conectar una vez y mantener la conexión viva para todas las solicitudes siguientes, lo que da lugar a un alto rendimiento. Por eso, FastCGI se considera una alternativa al CGI de alto rendimiento a la hora de escribir aplicaciones para un servidor Web en distintos lenguajes, incluidos Perl, C, C++, Java y Python.

La existencia de CGI, FastCGI y de Server API crea una gran confusión en los desarrolladores y en los administradores de sistemas. Para aclarar estos conceptos, la tabla 14.1 proporciona una comparación entre estas tecnologías con respecto a una serie de características claves.

**Tabla 14.1.** Comparar CGI, Server API y FastCGI

Característica	CGI	Server API	FastCGI
Dependencia del lenguaje de programación	Es independiente del lenguaje. Las aplicaciones CGI se pueden escribir en prácticamente cualquier lenguaje de programación (normalmente C/C++).	Las aplicaciones se tienen que escribir en un lenguaje que soporte el lenguaje de programación del fabricante del API.	Es independiente del lenguaje. Al igual que ocurre con CGI, las aplicaciones FastCGI se pueden escribir en cualquier lenguaje de programación.
Aislamiento del proceso	Procesos separados para las aplicaciones; las aplicaciones de depuración no pueden hacer quebrar el servidor Web o acceder al estado interno privado y comprometer la seguridad. Los errores en el servidor central pueden corromper las aplicaciones.	No soporta esta característica. Como estas aplicaciones se ejecutan en el espacio de la dirección del servidor, las aplicaciones de depuración pueden corromper el servidor principal.	Una aplicación de depuración no puede quebrar o corromper el servidor central ni otras aplicaciones.
Tipo de estándar	Estándar abierto. En cada servidor Web se implementa algún formulario CGI.	Propietario. Al codificar su aplicación con un API determinado, le bloquea en un servidor de un fabricante particular.	No propietario, propuesto para estándar abierto. El soporte se encuentra bajo el desarrollo de otros servidores Web, incluidos servidores comerciales de Microsoft y Netscape. Apache actualmente soporta FastCGI como módulo de terceras partes.

Característica	CGI	Server API	FastCGI
Dependencia de plataforma	Independiente de plataforma. CGI no está atado a ninguna arquitectura de servidores (de un solo hilo, multihilo, y similares).	Se encuentra atada a la arquitectura del servidor. Las aplicaciones API tienen que compartir la misma arquitectura que el servidor. Si el servidor Web es multihilo, la aplicación tiene que ser asegurar los hilos. Si el servidor Web tiene procesos de un solo hilo, las aplicaciones multihilo no tendrán ventajas de rendimiento.	Independiente de plataforma. El FastCGI no está atado a ninguna arquitectura de servidor. Cualquier servidor Web puede implementar la interfaz FastCGI.
Rendimiento	Se crea un nuevo proceso para cada solicitud y la lanza si se realiza la solicitud; poca eficacia.	Las aplicaciones se ejecutan en el proceso del servidor y son persistentes a lo largo de las solicitudes. No existe el problema del arranque e iniciación del CGI.	Los procesos FastCGI son persistentes; se reutilizan para manejar varias solicitudes. No existe el problema del arranque e iniciación del CGI.
Complejidad	Fácil de entender.	Muy complicado. Los fabricantes de las API introducen una curva de aprendizaje, con aumento de implementación de costes de mantenimiento.	Sencillo, con migración fácil desde CGI.
Arquitectura distribuida	No soporta. Para ejecutar aplicaciones CGI en un sistema remoto, se necesita un servidor Web en ese sistema, porque los servidores Web ejecutan las aplicaciones.	Depende del fabricante.	Soportada. Las aplicaciones FastCGI se pueden ejecutar en cualquier host que soporte TCP/IP.

## Alcanzar alto rendimiento utilizando caching

¿Cómo es FastCGI? La respuesta depende de la aplicación. Si una aplicación lee datos desde archivos y puede cachearlos en la memoria, la versión FastCGI de esta aplicación proporciona mayor rendimiento que una aplicación CGI o una aplicación basada en API. Una aplicación CGI no puede utilizar la memoria caché porque se ejecuta una nueva instancia de la aplicación por cada solicitud y abandona una vez que finaliza el proceso. Del mismo modo, las aplicaciones API

se ejecutan en procesos hijo que no comparten memoria, y por lo tanto, no se puede aplicar el cacheamiento de datos. Incluso si se implementa el cacheamiento en memoria para cada proceso hijo en este modelo, funciona mal porque cada proceso hijo tiene que tener una copia del caché en la memoria, lo que gasta una gran cantidad de memoria.

FastCGI está diseñado para permitir un cacheamiento en memoria eficaz. Las solicitudes se dirigen desde cualquier proceso hijo al servidor de la aplicación FastCGI. El proceso de la aplicación FastCGI mantiene un caché en memoria. Observe que, en algunos casos, una sola aplicación FastCGI puede no proporcionar suficiente rendimiento. Con procesos multihilo ejecuta un proceso de aplicación diseñado para manejar varias solicitudes al mismo tiempo. Los hilos manejan solicitudes concurrentes compartiendo memoria de proceso, de modo que todas tienen el mismo caché.

## Escalabilidad a través de aplicaciones distribuidas

A diferencia de lo que ocurre con las aplicaciones CGI, las aplicaciones FastCGI no obtienen las variables de entorno CGI de la tabla de procesos. Lo que hacen es realizar una conexión full-duplex entre la aplicación y el servidor Web que se utiliza para comunicar la información de entorno, las salidas y entradas estándares y los errores. Esto permite que las aplicaciones FastCGI se ejecuten en máquinas remotas utilizando conexiones TCP/IP con el servidor Web, tal y como se muestra en la figura 14.1. Esta figura muestra que las solicitudes desde Internet se manejan con [www.nitec.com](http://www.nitec.com) (el servidor Web), que conecta de forma remota mediante una conexión TCP con [fcgi.nitec.com](http://fcgi.nitec.com) donde los scripts FastCGI manejan las solicitudes.

### El ritmo de FastCGI

Los desarrolladores de FastCGI realizaron pruebas utilizando tres versiones de una aplicación (basadas en CGI, FastCGI y en una especificación API basada en servidores Web muy conocida) que interaccionaban con un servidor de una base de datos. Los desarrolladores vieron que cuando la versión FastCGI de la aplicación utilizaba caching en memoria y conexiones persistentes con la base de datos, superaba con mucho el rendimiento de las versiones CGI y de las aplicaciones basadas en API.

Cuando se desactivaba el cache en memoria en la aplicación FastCGI, y se utilizaba una conexión persistente para la aplicación basada en API, la aplicación basada en API tenía un rendimiento ligeramente superior que la versión FastCGI. Esto significa que solo cuando se utiliza el mayor básico (es decir, cuando se desactivan ventajas de FastCGI como la característica

de caching en memoria) gana la versión API. ¿Pero por qué vamos a desactivar el caching? En otras palabras, a no ser que escriba una versión incompleta de una aplicación FastCGI, es probable que supere el rendimiento de las versiones CGI y API.

Las pruebas demostraron que las ventajas resultantes de la arquitectura de una aplicación basada en FastCGI daban lugar a un rendimiento tres veces superior que su equivalente API. Este factor es casi más alto si las aplicaciones tienen que conectarse con recursos remotos, como en el caso de un servidor de bases de datos remoto. Sin embargo también indican que un servidor Web multihilo capaz de mantener caché y conexiones persistentes para sus hilos de aplicación, tiene prácticamente el mismo rendimiento que las aplicaciones FastCGI. Esto se debe a la ausencia de sobrecarga de inter procesos de comunicación en un entorno de hilos. Desarrollar aplicaciones multihilo requiere un diseño y una programación muy cuidadosos, un fallo en un solo hilo puede hacer que se caiga todo el sistema del servidor Web. Por otro lado, los procesos FastCGI sacan partido del modelo de aislamiento de procesos, en el que se ejecutan como procesos externos. Esto proporciona una red segura para el sistema del servidor Web. En el caso de un fallo en una aplicación FastCGI, el servidor Web seguirá funcionando. Si le gustan los procesos multihilo puede escribir sus aplicaciones FastCGI en un modelo multihilo, que sigue teniendo las ventajas del modelo de procesos aislados.

[www.nitec.com](http://www.nitec.com)

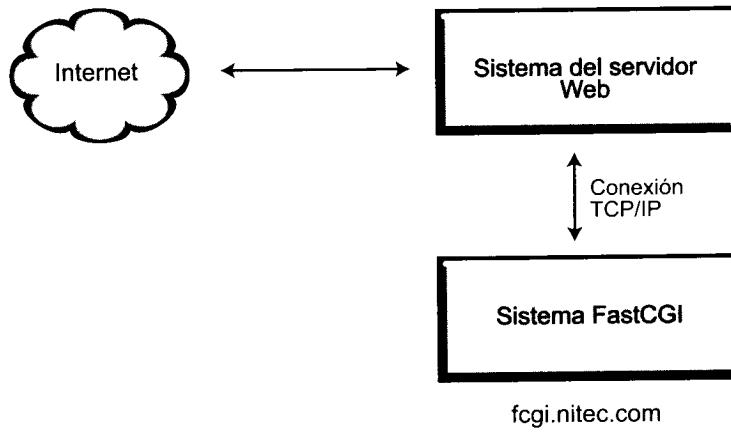


Figura 14.1. FastCGI en una máquina remota

Cuando las aplicaciones basadas en CGI y en API se convierten en cuellos de botella de rendimiento debido a una gran carga, la típica solución es un servidor Web más potente o más servidores Web. Utilizando aplicaciones FastCGI se puede ejecutar la carga en servidores de aplicación dedicados en la red, de modo

que liberan al servidor Web para lo que sabe hacer mejor, servir solicitudes Web. El servidor o los servidores Web pueden ser adaptados para realizar mejores servicios Web y, al mismo tiempo, el servidor de la aplicación FastCGI puede ser adaptado para ejecutar aplicaciones de forma eficaz. El administrador Web nunca tendrá que preocuparse de cómo equilibrar los requisitos de recursos del servidor Web y las aplicaciones en la misma máquina. Esto proporciona una configuración más flexible en el lado del servidor Web al tiempo que del lado de la aplicación.

Muchas organizaciones quieren proporcionar acceso a bases de datos en sus sitios Web. Debido a las limitaciones del CGI y de los API, sin embargo, deben replicar una versión limitada de la base de datos en el servidor Web para proporcionar este servicio. Esto da lugar a un trabajo considerable para el administrador. Con FastCGI remoto, las aplicaciones se pueden ejecutar en la red interna, simplificando el trabajo del administrador. Cuando se utiliza con una configuración apropiada del firewall y bajo una auditoría, esta aproximación proporciona un modo de ofrecer datos y aplicaciones internas a Internet seguro, de alto rendimiento y escalable.

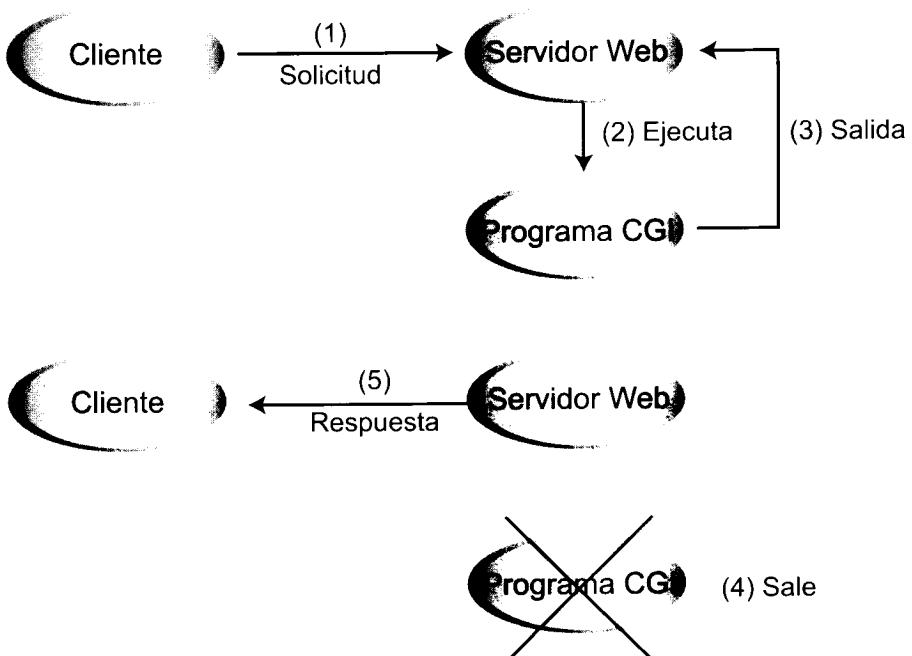
Las conexiones FastCGI remotas tienen dos procesos de seguridad: autenticación y privacidad. Las aplicaciones FastCGI deberían aceptar únicamente conexiones desde servidores Web en los que confiasen (la librería de aplicaciones incluye soporte para validación de direcciones IP). Las futuras versiones del protocolo podrían incluir soporte para aplicaciones de autenticación de servidores Web, así como para ejecutar conexiones remotas con protocolos de seguridad como Secured Socket Layer (SSL).

## Entender cómo funciona FastCGI

Las aplicaciones FastCGI utilizan una sola conexión para comunicarse con un servidor Web. La conexión se utiliza para enviar las variables de entorno y los datos STDIN, y las aplicaciones y los datos STDOUT y STDERR al servidor Web. La utilización de este sencillo protocolo de comunicación también permite que las aplicaciones FastCGI residan en una máquina distinta (o en distintas máquinas) que el servidor Web, permitiendo que las aplicaciones se encuentren en más de un solo sistema y proporcionando una integración sencilla con los sistemas existentes. Para aplicaciones locales, el servidor utiliza una tubería full-duplex para conectar con el proceso de la aplicación FastCGI. Para aplicaciones remotas, el servidor utiliza una conexión TCP/IP.

El protocolo FastCGI se utiliza para la comunicación entre el servidor Web y las aplicaciones empleando un formato de registro de paquetes. La mayoría de los desarrolladores utilizarán la librería de aplicaciones FastCGI y no tendrán que preocuparse por los detalles de protocolo. Sin embargo, las aplicaciones especializadas pueden implementar directamente el protocolo FastCGI.

Como el CGI es muy parecido al FastCGI, vamos a revisar el proceso de solicitudes CGI. La figura 14.2 muestra el modelo simplificado del procesamiento de solicitudes con CGI.

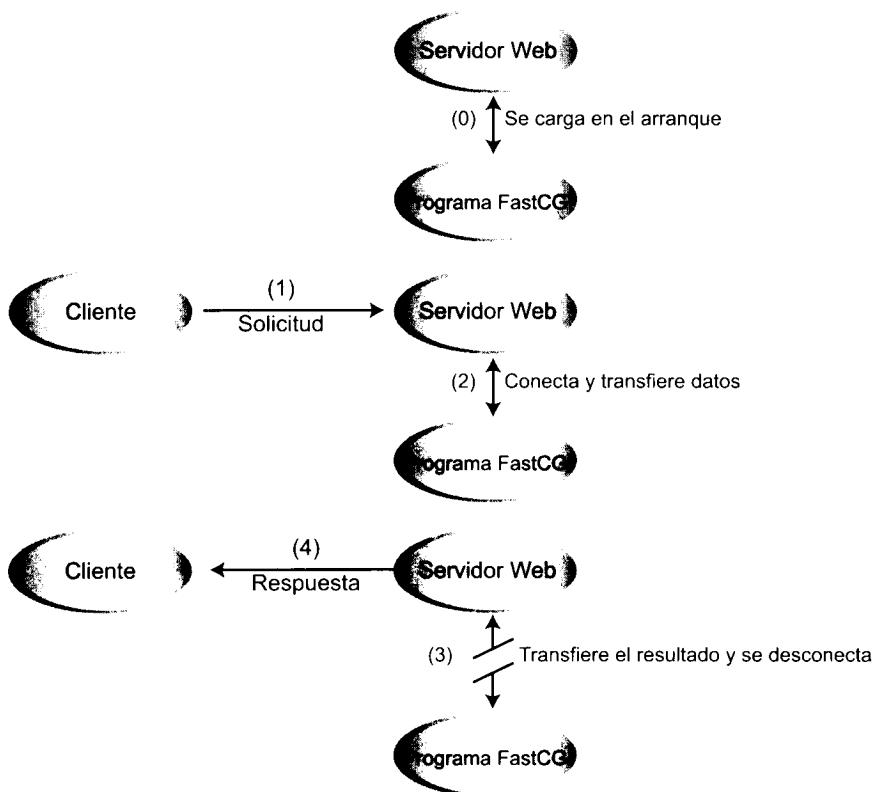


**Figura 14.2.** Modelo de procesamiento de una solicitud CGI

Para cada solicitud CGI, tiene lugar lo siguiente (referido a la figura):

1. El sistema del cliente envía una solicitud al servidor Web. El servidor Web determina si esta solicitud tiene que ser servida por un programa CGI o no.
2. El servidor Web crea un nuevo proceso CGI y el proceso se inicia. El servidor Web pasa información relacionada con la solicitud al programa, mediante variables de entorno. Dependiendo del método de solicitud (GET o POST), los datos del usuario se almacenan en una variable de entorno llamada `QUERY_STRING` o se colocan en la salida estándar del proceso.
3. La aplicación CGI realiza sus tareas y envía todas sus salidas a la salida estándar, que el servidor Web lee y analiza (con la excepción de las aplicaciones con cabeceras no analizadas).
4. El programa CGI sale y el servidor devuelve el resultado CGI al cliente.
5. El resultado del programa CGI se envía al sistema del cliente.

Los procesos FastCGI son persistentes. Cuando termina una solicitud, esperan otra en lugar de abandonar, tal y como muestra la figura 14.3.



**Figura 14.3.** Modelo de procesamiento de una solicitud con FastCGI

En el caso de aplicaciones con la cabecera sin analizar, la aplicación CGI es responsable de producir las cabeceras HTTP apropiadas, y en el resto de los casos el servidor Web produce las cabeceras HTTP apropiadas basándose en el tipo de contenido encontrado en el STDOUT del programa. El servidor Web registra cualquier información sobre errores que se escriba en el error estándar del programa CGI.

El servidor Web crea procesos de una aplicación FastCGI para manejar las solicitudes. Los procesos se podrían crear en el arranque o a demanda. El programa FastCGI se inicia a sí mismo y espera una nueva conexión desde el servidor Web. El procesamiento de solicitudes del cliente en una aplicación FastCGI de un solo hilo, funciona del siguiente modo:

1. Cuando entra una solicitud del cliente, el servidor Web decide si la conexión tiene que ser manejada con un programa FastCGI o no.
2. Si la solicitud tiene que ser servida por un programa FastCGI, el servidor Web abre una conexión a los procesos FastCGI, que ya se están ejecutando.
3. El servidor envía la información de la variable de entorno CGI y realiza una salida estándar a la conexión. El proceso FastCGI envía la informa-

- ción de la salida estándar y de los errores de vuelta al servidor por esa misma conexión, entonces el proceso FastCGI cierra la conexión.
4. El servidor Web responde al cliente con el dato que ha enviado el proceso FastCGI, completando la solicitud. El proceso FastCGI espera entonces otra conexión del servidor Web.

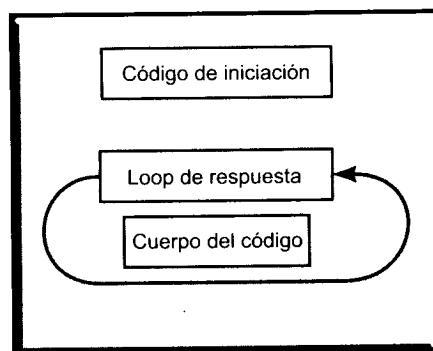
## Arquitectura básica de una aplicación FastCGI

Como ya sabe, a diferencia de un programa CGI, un programa FastCGI mantiene la ejecución después de procesar la solicitud. Esto le permite procesar futuras solicitudes tan pronto como llegan, y además hace la arquitectura del programa FastCGI distinta de la arquitectura de un programa CGI. Un programa CGI ejecuta de forma secuencial y abandona, mientras que un programa FastCGI ejecuta de forma secuencial y realiza un ciclo continuo. La figura 14.4 muestra la arquitectura básica de una aplicación FastCGI.

Tal y como muestra la figura, un programa FastCGI tiene un segmento de código de inicialización y un segmento con un loop de respuesta que encapsula el cuerpo del programa. El código de inicio se ejecuta una vez, cuando se inicia la aplicación. El código de inicio normalmente realiza operaciones que consumen mucho tiempo como el abrir una base de datos o calcular valores para tablas.

El loop de respuesta se ejecuta en continuo, espera que llegue una solicitud del cliente. El loop empieza con una llamada a `FCGI_Accept`, una rutina que se encuentra en la librería FastCGI. La rutina `FCGI_Accept` bloquea la ejecución del programa hasta que un cliente solicita la aplicación FastCGI. Cuando llega la solicitud de cliente, `FCGI_Accept` lo desbloquea, ejecuta una iteración sobre el loop de respuesta, y entonces lo vuelve a bloquear, esperando otra solicitud de un cliente. El loop finaliza únicamente cuando el administrador del sistema o el servidor Web asesina la aplicación FastCGI.

Arquitectura básica de una aplicación FastCGI



**Figura 14.4.** Arquitectura básica de una aplicación FastCGI

Se ejecuta el cuerpo del programa en cada iteración del loop de respuesta. En otras palabras, el cuerpo se ejecuta una sola vez para cada solicitud. FastCGI determina la información de la solicitud, como variables de entorno y datos de entrada, antes de cada iteración en el código del cuerpo. Cuando el código del cuerpo se ejecuta, una llamada a `FCGI_Accept` le dice al servidor que este programa ha completado una solicitud y que está preparado para recibir otra. En este momento `FCGI_Accept` bloquea la ejecución hasta que recibe una nueva solicitud.

Las aplicaciones FastCGI pueden tener un solo hilo de ejecución o varios. Para aplicaciones de un solo hilo, el servidor Web mantiene un grupo de procesos FastCGI (si la aplicación se está ejecutando localmente) para manejar solicitudes de clientes. El tamaño del grupo es configurable por parte del usuario. Las aplicaciones FastCGI de varios hilos de ejecución, pueden aceptar varias conexiones del servidor Web y pueden manejarlas simultáneamente en un solo proceso.

## Distintos tipos de aplicaciones FastCGI

Otro aspecto importante de FastCGI es que soporta roles (tipos) de aplicaciones. A diferencia de una aplicación CGI, una aplicación FastCGI es persistente, y por lo tanto puede utilizarse para propósitos diferentes que las aplicaciones basadas en CGI. Los siguientes párrafos discuten dos tipos nuevos de aplicaciones que puede soportar FastCGI.

Una aplicación FastCGI puede hacer todo lo que hace una aplicación CGI, de modo que la típica aplicación FastCGI es igual que su equivalente en CGI. La siguiente lista muestra los roles de aplicación disponibles con el soporte FastCGI:

- **Filtros:** puede crear una aplicación filtro FastCGI para procesar un archivo solicitado antes de devolverlo al cliente. Por ejemplo, imagine que quiere aplicar un conjunto estándar de cabeceras y pies de página para cada página HTML (.html) devuelta por el servidor Web. Esto es posible utilizando una aplicación de filtro FastCGI. Cuando llega una solicitud de un archivo .html al servidor, envía el archivo solicitado al filtro FastCGI responsable de añadir la cabecera y el pie de página. La aplicación FastCGI devuelve la página HTML resultante al servidor, el cual la transmite al cliente. Las aplicaciones de filtro FastCGI pueden suponer un importante incremento del rendimiento mediante el caching de los resultados filtrados (el servidor proporciona el momento de modificación en la información solicitada, de modo que la aplicación puede descargar el caché cuando no se modifica el archivo). Las aplicaciones de filtro son útiles en el desarrollo de analizadores de páginas HTML con sentencias SQL embebidas, en los convertidores de formato al vuelo, y similares.
- **Aplicaciones de autenticación externas:** otro tipo nuevo de aplicaciones que se puede desarrollar utilizando soporte FastCGI incluye la autentifica-

ción externa de programas y gateways para aplicaciones de autentificación de terceras partes. Por ejemplo, si utiliza un servidor externo de bases de datos para almacenar información de autentificación como nombre de usuario, contraseña u otros datos específicos de permisos, puede crear una aplicación FastCGI para mantener una conexión persistente con el servidor de la base de datos y realizar consultas para autenticar las solicitudes de acceso. ¿Se puede hacer esto con una aplicación CGI? Sí, excepto que una aplicación CGI tiene que abrir la conexión con el servidor de la base de datos cada vez que se ejecuta. Esto puede resultar muy caro en términos de utilización de recursos (CPU, red).

Por otro lado, la versión FastCGI de la misma aplicación mantiene una sola conexión con el servidor de la base de datos, realiza las consultas, y devuelve el código de estado HTTP apropiado basado en los resultados de las consultas. Por ejemplo, cuando una solicitud de acceso se acompaña de un par válido nombre de usuario/contraseña, la aplicación FastCGI consulta al servidor de la base de datos para determinar si el par tiene el acceso permitido al recurso solicitado. Si la base de datos devuelve un valor específico indicando que el acceso debería permitirse, la aplicación FastCGI devuelve un código de estado HTTP 200 OK; cuando falla la autorización, puede enviar un código de estado HTTP distinto, como 401 Unauthorized.

## Migración desde CGI a FastCGI

Una de las principales ventajas del FastCGI es que el camino de migración de CGI a FastCGI es razonablemente sencillo. Los scripts CGI especiales basados en Perl, que utilizan el módulo CGI.pm, se pueden convertir fácilmente en aplicaciones FastCGI.

Cualquier programa CGI escrito en otros lenguajes como C, C++, Tcl o Java, también se pueden convertir utilizando el Software Development Kit (SDK) de FastCGI.

**CD-ROM:** Los desarrolladores de las especificaciones de FastCGI proporcionan un kit de software de desarrollo (SDK) disponible gratuitamente para facilitar el proceso del desarrollo de una aplicación FastCGI. El SDK está incluido también en este CD-ROM. Este kit, es un archivo tar comprimido, que le ayuda a escribir aplicaciones FastCGI en C, C++, Perl, Tcl y en Java. Cuando descomprime y extrae el archivo tar, crea un directorio fcgi-devel-kit. Un archivo index.html proporciona información sobre lo que hay disponible en el kit. El kit se puede obtener también desde el sitio Web de FastCGI en [www.fastcgi.com/applibs](http://www.fastcgi.com/applibs).

# Puntos que hay que recordar sobre la migración

La siguiente lista le da una serie de trucos a recordar cuando realizamos una migración desde una aplicación CGI:

- Hay que tener cuidado cuando una aplicación CGI que ha migrado, tiene código que puede interferir con una segunda ejecución del código del cuerpo, es necesario arreglar esto. La solución a este problema podría ser tan sencilla como añadir código para reiniciar algunas variables, arrays, y similares. La aplicación debe asegurar que cualquier estado que se crea en el proceso de una solicitud, tiene un efecto intencionado en solicitudes posteriores.
- Hay una práctica común entre los desarrolladores CGI, que es subdividir una gran aplicación en pequeños applets CGI, para compensar la desventaja asociada a la inicialización, con las aplicaciones CGI. Con FastCGI, es mejor tener la funcionalidad relacionada en un solo ejecutable de modo que haya menos procesos que manejar y las aplicaciones puedan sacar partido de la compartición de la información cacheada entre las funciones.
- Para migrar con facilidad a FastCGI, los ejecutables construidos con el módulo FCGI pueden ejecutar programas CGI o programas FastCGI, dependiendo de cómo se invoquen. El módulo detecta el entorno de ejecución y automáticamente selecciona FastCGI o rutinas I/O, dependiendo de lo que sea apropiado.
- Muchas aplicaciones CGI están escritas de modo que no puedan intentar realizar ninguna operación en la memoria. Esto es consecuencia de aplicaciones CGI que existen tras la ejecución. Y en la mayoría de los casos, el sistema operativo es capaz de restablecer la memoria para otros usos. Por encima de esto, muchas aplicaciones CGI ni siquiera intentan cerrar, ya que la responsabilidad se pasa al sistema operativo.

En este caso, es muy importante que este tipo de aplicaciones se fijen durante la migración a la versión FastCGI. Recuerde que las aplicaciones FastCGI residen en memoria hasta que el servidor Web o el administrador las eliminan. Si una aplicación CGI que filtra memoria, se convierte en FastCGI sin nadie que se ocupe del problema, la versión FastCGI podría seguir filtrando memoria, causando a la larga un fallo de recursos. Evite largos fines de semana en la oficina solucionando con anticipación el problema. Si la aplicación CGI es muy complicada, y fijarla adecuadamente resulta muy caro en términos de tiempo y esfuerzo, hay otra solución disponible.

Puede mantener la cuenta de cuántas veces esta aplicación de filtrado de memoria ha servido solicitudes y las ha eliminado. El listado 14.1 muestra un sencillo ejemplo de cómo se mantiene una cuenta de procesamiento de solicitudes en

una aplicación FastCGI basada en C. Como puede ver en el listado, cuando se procesa el número máximo de solicitudes, la aplicación FastCGI abandona.

#### Listado 14.1. memory\_hog.c

```
#include <fcgi_stdio.h>
void main(void){
    int maxRequests = 100;
    // número máximo de solicitudes antes de salir.
    int requestCount = 0;
    // se utiliza para mantener la cuenta de las solicitudes
    // procesadas
    // hasta ahora.

    while(FCGI_Accept() >= 0) {
        // inicio del loop de respuesta
        /* código del cuerpo */

        printf("Content-type: text/html\r\n");
        printf("\r\n");
        printf("Number of requests processed so far is %d.", requestCount++);
        /* código de filtrado de memoria */

        /* fin del código de aplicación */

        if(requestCount >= maxRequests) {
            /* código de limpieza */
            exit 0;
        }
        /* final del código del cuerpo */
        .
    }
    /* final o código de limpieza (si existe) */
    exit(0);
}
```

## Un ejemplo de un script de migración

Esta sección demuestra cómo migrar `fontsize.cgi`, un script CGI script sencillo (escrito en Perl), que se encuentra en el listado 14.2.

#### Listado 14.2. fontsize.cgi

```
#!/usr/bin/perl -w
# Variables
```

```

my $MAX = 10;
my $i;

# Cabecera de contenido
print "Content-type: text/html\n\n";

# loop principal
for ($i=0; $i < $MAX; $i++){

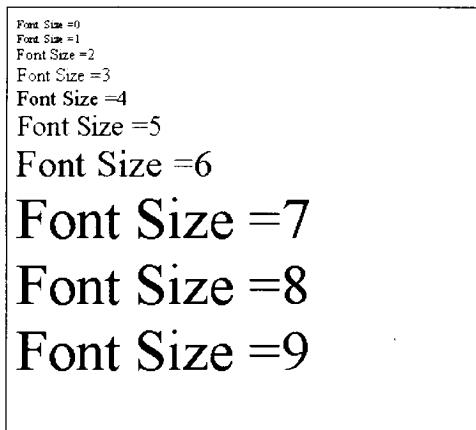
    print "<FONT SIZE=$i>Font Size = $i</FONT><BR>";

}

exit 0;

```

Esta aplicación CGI produce la salida de la figura 14.5.



**Figura 14.5.** Salida del script fontsize.cgi

Ahora vamos a convertir esto en una aplicación FastCGI. Lo primero que tiene que hacer es asegurarse de que tiene instalado el módulo FCGI desde el sitio Comprehensive Perl Archive Network (CPAN) para seguir los pasos siguientes:

1. Como raíz, desde la línea de comando ejecute el comando siguiente:  
`perl -MCPAN -e shell`
  2. Desde el prompt `cpan>` introduzca el comando `install FCGI`, que instalará la última versión de este módulo CPAN, si no la tiene.
  3. Como segunda línea del script, añada la siguiente:  
`use FCGI;`
- Esta línea le dice a Perl que cargue el módulo FCGI cuando ejecute el script.
4. Identifique el bloque de inicio de la aplicación CGI. Para este script, el bloque de inicio consiste en dos declaraciones de variables:

```
my $MAX = 10;
my $i;
```

5. Identifique los bloques del loop de respuesta y del cuerpo de la aplicación CGI. Como cada solicitud tiene que responderse con una cabecera Content-Type adecuada, el cuerpo del código principal empieza con la línea siguiente:

```
print "Content-type: text/html\n\n";
```

El resto del cuerpo es para el loop responsable del verdadero proceso.

6. Ahora que el código del cuerpo está identificado, tiene que poner el loop de respuesta a su alrededor. Esto se lleva a cabo utilizando el código siguiente:

```
while(FCGIaccept() >= 0) {
    # Código del cuerpo
}
```

Ahora el cuerpo es el siguiente:

```
while(FCGI::accept() >= 0) {
    print "Content-type: text/html\n\n";

    for ($i=0; $i < $MAX; $i++){
        print "<FONT SIZE=$i>Font Size = $i</FONT><BR>";
    }
}
```

Las aplicaciones FastCGI actúan como las aplicaciones de un servidor, de modo que la línea siguiente sólo se ejecuta si el servidor Web o el administrador envían una señal para finalizar:

```
exit 0;
```

La versión FastCGI de la aplicación completa se encuentra en el listado 14.3.

#### Listado 14.3. fontsize.fcgi

```
#!/usr/bin/perl

# Carga el módulo FCGI
use FCGI;

# variables globales
my $MAX = 10;
my $i;

# Loop FastCGI
```

```

while(FCGI::accept() >= 0) {

    # Cabecera de contenido
    print "Content-type: text/html\n\n";

    # Loop principal
    for ($i=0; $i < $MAX; $i++) {
        print "<FONT SIZE=$i>Font Size = $i</FONT><BR>";
    }
}

exit 0;

```

## Establecer FastCGI en Apache

Para instalar `mod_fastcgi` con Apache, siga los pasos siguientes:

1. Baje la última versión del módulo `mod_fastcgi` para Apache de [www.fastcgi.com](http://www.fastcgi.com).
2. Extraiga la fuente en el directorio `/usr/local/src`. Se crea un nuevo directorio llamado `mod_fastcgi_version`. Lea el archivo `INSTALL` en este nuevo directorio para obtener los detalles de la instalación de este módulo, ya que la instalación varía cada cierto tiempo.
3. Copie el nuevo directorio en un subdirectorio llamado `fastcgi` bajo el directorio de módulos dentro del árbol de la fuente Apache.
4. Desde el directorio de la distribución fuente de Apache, ejecute `configure --enable-module=fastcgi` y cualquier otra opción que quiera especificar con el script `configure`. (Ver el capítulo 3 para obtener los detalles de otras opciones disponibles para el script `configure`.) Si ya tiene compilado Apache puede reemplazar `configure` con `config.status` en el comando anterior para asegurarse de que todas las opciones que especificó, cuando ejecutó `configure` la última vez, se han aplicado de nuevo.
5. Ejecute `make && make install` para compilar e instalar Apache con soporte `mod_fastcgi`.
6. Reinicie Apache utilizando el comando `/usr/local/apache/bin/apachectl`.

## Directivas FastCGI para Apache

El módulo `mod_fastcgi.c` proporciona un conjunto de directivas para controlar la utilización de las aplicaciones FastCGI. Estas directivas se discuten a continuación:

## Directiva AppClass

Esta directiva le permite iniciar aplicaciones FastCGI.

**Sintaxis:** AppClass path\_to\_FastCGI\_application [-processes N] [-listen-queue-depth N] [-restart-delay N] [-priority N] [-port N] [-socket sock-name] [-initial-env key=value] [-restart-delay N] [-priority N] [-port N] [-socket sock-name] [-initial-env key=value]

**Contexto:** configuración del servidor

Por ejemplo, la siguiente directiva permite a mod\_fastcgi cargar /www/development/fcgi-devel-kit/echo.fcg FastCGI application\:

```
AppClass /www/development/fcgi-devel-kit/echo.fcg -port 9000
```

La aplicación escucha el puerto 9000. Cuando muere una aplicación FastCGI que se ha cargado con esta directiva, mod\_fastcgi reinicia la aplicación y escribe una entrada de registro en el archivo de registro de errores. Los parámetros opcionales para la directiva AppClass son:

- **processes:** esta opción determina el número de procesos FastCGI que se crean. En un escenario de alta carga, la carga de muchas instancias de la misma aplicación FastCGI puede proporcionar mejor rendimiento. El valor por defecto es 1.
- **listen-queue-depth:** esta opción determina la profundidad de la cola de escucha para la aplicación o las aplicaciones FastCGI cargadas con esta directiva. El valor por defecto es suficiente en la mayoría de los casos, pero en escenarios con alta carga, puede aumentar la profundidad de la cola. Esto disminuye la posibilidad de que una respuesta se rechace porque la aplicación está muy ocupada. Sin embargo, si espera una gran carga, y su servidor es capaz de ejecutar algunos procesos FastCGI extra, aumente el número de procesos en lugar de la profundidad de la cola de escucha. El valor por defecto es 5.
- **restart-delay:** esta opción determina el número de segundos que se utilizarán para retrasar un reinicio de un proceso FastCGI muerto. Esto sólo es útil cuando está utilizando varias instancias de la misma aplicación FastCGI. En el caso de una sola aplicación FastCGI, se reinicia inmediatamente, y esta opción no tiene efecto. El valor por defecto es 5. Se debería preguntar por qué se necesita este retraso. Normalmente una aplicación FastCGI no debería morir; si muere, es que algo va mal. En este caso, el retraso le permite al servidor realizar otras tareas de utilidad además de reiniciar repetidamente la aplicación fallida.

- **Priority:** esta opción determina la prioridad de procesos de una aplicación FastCGI. El valor por defecto le permite a la aplicación FastCGI tener la misma prioridad que el servidor Apache. La llamada al sistema `setpriority` del sistema operativo define otros valores apropiados. En un sistema Red Hat Linux, la llamada al sistema `setpriority` permite un valor entre -20 y 20. Cuanto menor es el número, más favorable es el esquema utilizado para el proceso. Sin embargo, `mod_fastcgi` no permite valores negativos, lo que significa que no puede establecer que un proceso FastCGI tenga mayor prioridad que el servidor Apache. Por eso, todo lo que puede hacer es utilizar un número entero positivo para reducir la prioridad de la aplicación. El valor por defecto es el 0.
- **Port:** esta opción determina el puerto TCP que escuchará la aplicación FastCGI. Como los puertos por debajo de 1024 se utilizan para servicios estándar, tiene que utilizar un número de puerto superior. La utilización de esta opción le permite acceder a la aplicación desde otro sistema. No se proporciona ninguna variable de entorno por defecto.
- **Socket:** esta opción determina el nombre de la ruta del socket del dominio Unix que escucha la aplicación. El módulo crea este socket dentro del directorio especificado por la directiva `FastCgiIpcDir`. El valor por defecto es `Default Socket`. Si no proporciona la opción `port` ni la opción `socket`, el módulo crea un socket del dominio Unix para la aplicación.
- **initial-env:** esta opción se puede utilizar para insertar una variable de entorno en la tabla de entorno de la aplicación FastCGI. Puede utilizar esta opción varias veces para insertar más de un par clave = valor en el entorno de la aplicación. No se proporciona ninguna variable de entorno por defecto.

Observe que las opciones `-socket` y `-port` se excluyen mutuamente. Los nombres de las rutas no deben ser iguales que el nombre suministrado de la ruta en una directiva `AppClass` o `ExternalAppClass` anterior.

## **Directiva ExternalAppClass**

Utilice esta directiva cuando tenga una aplicación FastCGI ejecutándose en otro sistema.

**Sintaxis:** `ExternalAppClass FastCGI_application_name [-host host:port] [-socket sock_name]`

**Contexto:** configuración del servidor

En el siguiente ejemplo, la aplicación `echo.fcg` se ejecuta en un host llamado `fcgi.nitec.com` y escucha el puerto 9090. `<FastCGI-application-name>` es simplemente un identificador que se puede utilizar para describir la

ejecución de la aplicación en el host remoto, de modo que puede ser el nombre que quiera. Asegúrese de que elige un nombre que no se está utilizando en otra directiva AppClass o ExternalAppClass.

```
ExternalAppClass echo.fcg -host fcgi.nitec.com:9090
```

Los parámetros opcionales para la directiva ExternalAppClass son:

- **host:** esta opción le permite especificar el host y el número de puerto TCP de la aplicación FastCGI que se está ejecutando en otro sistema. Utilice el formato host:puerto para especificar el host y el puerto. Puede utilizar también un nombre de host o una dirección IP.
- **socket:** esta opción le permite especificar el nombre de la ruta del socket del dominio Unix que está utilizando una aplicación FastCGI.

## Directiva FastCgiIpcDir

Esta directiva especifica la ruta por defecto para los socket del dominio Unix creados por el módulo. La localización por defecto /tmp es perfecta siempre que no tenga trabajos cron establecidos para limpieza de su directorio /tmp cada cierto tiempo.

**Sintaxis:** FastCgiIpcDir path\_to\_UNIX\_domain\_socket

**Predefinido:** FastCgiIpcDir /tmp

**Contexto:** configuración del servidor

El nombre del socket tiene el siguiente formato:

OM\_WS\_n.pid

n es un número y pid es el ID del proceso principal de Apache. Si no le asigna una ruta a esta directiva, asegúrese de que únicamente Apache tiene acceso de lectura y escritura en el directorio.

## Configurar httpd.conf para FastCGI

Para configurar aplicaciones FastCGI, tiene una combinación de directivas mod\_fastcgi y directivas proporcionadas por el servidor Apache.

Utilice la directiva AppClass para iniciar las aplicaciones FastCGI que quiera que maneje este servidor Web. Las aplicaciones se manejan en el sentido de que el servidor registra un mensaje de error cuando un proceso manejado muere e intenta reiniciar los procesos manejados que han muerto.

Utilice una o las dos directivas AppClass y ExternalAppClass para definir una asociación entre un nombre de ruta y la información sobre la conexión para una aplicación FastCGI. La información sobre la conexión es el nombre de la ruta de un socket del dominio Unix o la dirección IP y el número de puerto de

un puerto TCP. La diferencia entre las dos directivas es que una sola directiva AppClass inicia una aplicación y establece la asociación para comunicarse con ella, mientras que ExternalAppClass sólo define la asociación. En el caso de AppClass, el nombre de la ruta utilizado en la asociación siempre es el nombre de la ruta del ejecutable de la aplicación; con ExternalAppClass, el nombre de la ruta es arbitrario.

Para que mod\_fastcgi procese una solicitud HTTP, el manejador de solicitudes debe ser fastcgi-script y el tipo MIME de las solicitudes deben ser application/x-`httpd-fcgi`. Apache proporciona varias formas de asignar el manejador y el tipo MIME para una solicitud:

- SetHandler (en el contexto de la sección de localización o directorio o en el archivo de control de acceso a directorios) puede asociar el manejador fastcgi-script con un archivo determinado o con todos los archivos de un directorio.
- AddHandler se puede asociar el manejador fastcgi-script con archivos basados en la extensión del archivo.
- ForceType (en el contexto de la sección de localización o directorio o en el archivo .htaccess) puede asociar el tipo MIME application/x-`httpd-fcgi` con un archivo determinado o con todos los archivos de un directorio.
- AddType se puede asociar el tipo MIME application/x-`httpd-fcgi` con archivos basados en una extensión de archivos.

Una vez que está configurado el módulo mod\_fastcgi, maneja solicitudes del siguiente modo:

1. Se recupera la información asociada con el nombre de la ruta solicitada. Si no hay asociada información con el nombre de la ruta, el servidor devuelve un código de error 404 Not Found.
2. El módulo mod\_fastcgi conecta con el proceso de la aplicación FastCGI. Si la conexión que se ha intentado falla, el servidor devuelve 500 Server Error.
3. La solicitud se transmite al proceso de la aplicación FastCGI, el cual genera una respuesta.
4. Se recibe la respuesta de la conexión y se transforma en una respuesta HTTP. El servidor envía esta respuesta de vuelta al cliente.

Debería respaldar su archivo httpd.conf, pero no utilice el archivo de ejemplo de configuración del listado 14.4 inmediatamente. Hay una configuración mínima que sólo utiliza el archivo httpd.conf con propósitos de prueba. Utilice esta configuración para pruebas iniciales con FastCGI, y cuando haya verifi-

cado que esta configuración funciona, fusione los aspectos específicos de FastCGI de esta configuración con su propia configuración.

**Listado 14.4. Ejemplo de un archivo de configuración httpd.conf**

```
# httpd.conf-minimal for mod_fastcgi

Port 80

# Deberia reemplazar la directiva User/Group con el nombre
# user/group apropiado
User $HTTP_USER
Group $HTTP_GROUP

# Configure sólo un proceso hijo desocupado,
# para simplificar la depuración
StartServers 1
MinSpareServers 1
MaxSpareServers 1

# digale a httpd donde deberia residir,
# activa el registro de acceso y de error
ServerRoot      $APACHE
ErrorLog        logs/error.log
TransferLog     logs/access.log
ScoreBoardFile   logs/httpd.scoreboard

# digale a http de donde obtener los documentos
#
DocumentRoot   $FASTCGI

# Este es el modo en el que deberia colocar los archivos socket
# Unix del dominio Unix en el directorio de registros (tendrá
# que crear un subdirectorio para ellos.)
# ;No lo haga a no ser que compruebe que todo funciona con los
# archivos socket almacenados localmente, en /tmp!
# FastCgiIpcDir $APACHE/logs
# Inicie el echo app
#
AppClass $FASTCGI/examples/echo -initial-env SOMETHING=NOTHING

# Tiene solicitudes manejadas por mod_fastcgi para este echo
# app (si no, ;el servidor devolverá el binario de app como un
# archivo!)
#
<Location /examples/echo>
    SetHandler fastcgi-script
</Location>

# Inicia una aplicación FastCGI que es accesible desde otras
# máquinas
AppClass $FastCGI/examples/echo.fcg -port 8978
```

```

<Location /examples/echo.fcg>
    SetHandler fastcgi-script
</Location>

# Conecta con el app "remote" iniciado antes. Como el app es
# realmente local, la comunicación tendrá lugar utilizando el
# loop de retorno TCP.
# Para comprobar la veracidad de la operación remota, inicie
# una copia de este servidor Web en otra máquina, e inicie otra
# copia con "localhost" de la linea con el nombre del host de
# la primera máquina.
#
# ExternalAppClass remote-echo -host localhost:8978

<Location /examples/remote-echo>
    SetHandler fastcgi-script
</Location>

# Debería manejar de este modo cualquier solicitud
# para un archivo con la extensión .fcg:
AddHandler fastcgi-script fcg

# Final de httpd.conf

```

Asegúrese de que tiene construido el nuevo `httpd` con el módulo `mod_fastcgi` y que el kit de desarrollo de FastCGI está construido adecuadamente. He utilizado el programa `echo` del kit de desarrollo de FastCGI en el directorio de ejemplos. No olvide reiniciar Apache cuando haya colocado el nuevo archivo `httpd.conf` del listado 14.4 en el directorio `$APACHE/conf`. Utilice un navegador para acceder a `http://$YOUR_HOST/examples/echo`.

`$YOUR_HOST` es la dirección IP del host ejecutando `httpd`. Busque `SOMETHING=NOTHING` en el entorno inicial que muestra `echo`. El contador de solicitudes debería aumentar cada vez que actualice la página. Antes de que pueda utilizar esta configuración, tiene que realizar algunas sustituciones, tal y como muestra la tabla 14.2.

**Tabla 14.2.** Sustituciones para la configuración de ejemplo

Palabra clave	Se reemplaza con
<code>\$APACHE</code>	Nombre completo de la ruta del directorio en el que se encuentra Apache.
<code>\$FASTCGI</code>	Nombre completo de la ruta del directorio en el que se encuentra su kit de desarrollo de FastCGI.
<code>\$HTTP_USER</code>	Nombre de usuario para la directiva <code>User</code> .
<code>\$HTTP_GROUP</code>	Nombre del grupo que utiliza para la directiva <code>Group</code> .





# **15 PHP y Apache**

---

## **En este capítulo**

1. Trabajamos con PHP.
2. Compilamos e instalamos PHP.
3. Configuramos PHP con el archivo `php.ini`.
4. Configuramos Apache para PHP.
5. Creamos un script PHP en línea de comandos.
6. Creamos scripts PHP.
7. Indexamos directorios con PHP.
8. Gestionamos páginas desaparecidas con PHP.
9. Manejamos sesiones de usuario con scripts PHP.
10. Accedemos a bases de datos MySQL con PHP.
11. Aseguramos archivos include PHP.
12. Autentificamos usuarios con PHP y con la base de datos MySQL.

Hace sólo unos años, PHP, el procesador de hipertexto PHP, comenzó a ser un macro lenguaje para HTML. Hoy se distribuye en un alto tanto por 100 del mundo de los sitios Web y es, de hecho, el método que hace de script del lado del cliente en las conexiones a las bases de datos. PHP es un lenguaje de programación en toda regla que se centra en aplicaciones Web dinámicas. Puede ejecutarse como un script independiente para realizar tareas de administración del sistema, de forma parecida a Perl y a otros lenguajes de programación shell, pero su objetivo principal son las páginas Web dinámicas que rinden datos desde una base de datos relacional como MySQL, Postgres, Oracle, DB2 y MS SQL Server.

Aunque PHP compite con ASP de Microsoft, Cold Fusion de Allaire, JSP de Sun, e incluso un primo de código fuente abierto llamado mod\_perl, realmente se encuentra por encima de prácticamente todos sus competidores por al menos un año. Como PHP es fácil de distribuir y de usar, tiene soporte en la mayoría de las plataformas y conectividad nativa a bases de datos, tiene el record más alto por ser fácil de aprender e implementar.

En este capítulo, aprenderá a compilar, a instalar y a configurar Apache para utilizar mod\_php, el módulo PHP para Apache. También aprenderá a asegurar PHP y a utilizar varias aplicaciones PHP.

Si está utilizando Apache en Windows, le mostraré cómo ejecutar scripts PHP en Apache para Windows en los siguientes capítulos.

## Entender cómo funciona PHP

PHP es un lenguaje interpretado que normalmente se utiliza para scripts del lado del servidor. Una página HTML se embebe con código PHP, que es analizado por un módulo del servidor o por un script CGI, para producir la salida deseada, que se envía al cliente Web; el cliente Web nunca ve el código PHP. A continuación tiene un ejemplo de una página con código PHP embebido.

```
<HTML>
<HEAD><TITLE> Simple PHP Example </TITLE></HEAD>
<BODY>

<?php echo "Welcome to PHP"; ?>

</BODY>
</HTML>
```

Esta página PHP imprime un string "Welcome to PHP" en el navegador Web una vez que lo analiza la ingeniería PHP del lado del servidor.

Este método en el que se embebe código dentro de una página HTML no está libre de controversias. A muchos desarrolladores de software conservadores no les gusta esta práctica porque mezcla código con presentación. Si considera HTML como un vehículo para la presentación de datos, entonces mezclar código en la

presentación resulta muy erróneo. Sin embargo, esto no les preocupa a millones de usuarios de PHP. Los que se oponen al PHP argumentan que mezclar código en la presentación dificulta el trabajo del experto en presentación. En el área de la Web, el experto en presentación es realmente un diseñador HTML o de gráficos que, probablemente, carece de conocimientos de programación. Por ese motivo, embeber scripts en páginas HTML haría la vida difícil a los diseñadores HTML. Sin embargo, PHP ofrece una gran variedad de librerías que funcionan con plantillas para resolver este problema, ofreciendo una metodología de desarrollo eficaz y un proceso sencillo de mantenimiento. De este modo, los desarrolladores se centran en la lógica de negocio, y los diseñadores de páginas pueden cambiar la distribución de una página dinámica sin afectar al desarrollo o interferir con el código PHP. Más de tres millones de sitios Web consideran que vale la pena utilizarlo, y han convertido PHP en una plataforma de desarrollo de aplicación rápida (Rapid Application –Development, RAD).

PHP se utiliza exclusivamente para el desarrollo Web. Se puede compilar también como un intérprete de script independiente, y puede manejar tareas sencillas de administración. La última versión de PHP (versión 4) tiene la ingeniería de lenguaje central, el analizador Zend, lo suficientemente abstracta como para que sea considerada por otras tecnologías, en las que PHP puede ser el lenguaje embedido en la plataforma como VBA se utiliza con Microsoft Office.

## PHP en su compañía

Aunque los software de código abierto como Linux, MySQL y Perl se están haciendo más habituales en los entornos corporativos, sigue siendo, a menudo, necesario vender este tipo de tecnología para gestiones de alto nivel. Cuando PHP cumple los requisitos, pero la gestión necesita una larga lista de razones, PHP puede ser la solución correcta, por los motivos siguientes:

- **Desarrollo Web rápido:** organizar y ejecutar scripts PHP es muy rápido. Escribir una aplicación de integración de una Web y una base de datos en PHP es una cuestión de horas o de días, mientras que los lenguajes de programación como C/C++ y Java pueden llevarnos meses. A diferencia de los lenguajes de propósito general, PHP es un lenguaje de script, con una afinidad particular por el desarrollo de páginas Web dinámicas.
- **Transportable a la mayoría de las plataformas de sistemas operativos:** PHP se ejecuta en todos los sistemas operativos importantes, incluidos Unix, Linux, Windows, OS/2 y Mac OS.
- **Transportable a la mayoría de las plataformas de servidores Web:** en PHP 4.0, la interfaz del servidor Web (Server API o SAPI) es abstracta, lo que le permite integrar perfectamente con servidores Web distintos, incluyendo Apache, IIS, iPlanet/Netscape Enterprise Server y Zeus.

- **Soporte integrado para la mayor parte de las bases de datos:** PHP implementa interfaces nativas para la mayor parte de las bases de datos importantes como MySQL, Postgres, Oracle, DB2 y MS SQL Server, lo que le hace un buen candidato para el desarrollo de Web gestionadas por bases de datos.
- **Alto rendimiento:** PHP 4 fue renovado para alto rendimiento; es varias veces más rápido que PHP 3. Un módulo gratuito llamado Zend Optimizer de Zend Technologies puede optimizar una aplicación y que ésta pase a ser entre 40 a 100 veces más rápida que cuando no estaba optimizada. Muchos sitios Web de gran tamaño utilizan PHP; puede leer sobre este módulo en el sitio Web de PHP Web, [www.php.net](http://www.php.net).
- **Fácil de aprender para desarrolladores nuevos:** PHP es como el lenguaje C, con gestión automática de memoria (es decir, no tiene que preocuparse por punteros). PHP tiene también elementos de Perl, Java y C++. Cualquiera que sepa alguno de estos lenguajes principales, puede aprender PHP en cuestión de horas o días.
- **Soporta estándares de Internet:** soporta estándares de Internet, como IMAP, FTP, POP, XML, WDDX, LDAP, NIS y SNMP. Esto significa que PHP puede funcionar de interfaz con distintos estándares y tecnologías con facilidad, todos ellos desde un conjunto de herramientas común, sin necesidad de módulos caros de terceras partes.
- **Amigo de las empresas:** muchas grandes empresas tienen comprada la plataforma Java ya y muchos amantes de Java encuentran la forma de hacer de Java la plataforma "ideal" para, incluso, el proyecto más pequeño, que a menudo da lugar a largos ciclos de desarrollo. Estas compañías suelen decir que no a cualquier cosa que indique otra lógica de negocio diferente a la implementada con Java y por lo tanto no hay cabida para PHP. PHP 4 soporta acceso directo a los objetos Java en muchos sistemas en los que tienen las máquinas virtuales de Java. Incluso soporta Distributed COM (DCOM) en plataformas Windows. PHP se está volviendo agresivo en cuanto a la integración con Java y con otras tecnologías de empresa de modo que está empezando a resultar complicado decir que no a PHP.
- **Comunidad de código fuente abierto:** el código fuente abierto ha resultado ser una aproximación de desarrollo de software muy eficaz. El PHP, como proyecto de código fuente abierto del grupo Apache, disfruta de los beneficios de esta aproximación. El grupo central de desarrolladores de PHP participa en muchos foros de soporte de código fuente abierto como son los grupos de noticias Usenet y los canales IRC. El soporte comercial también está disponible desde muchas compañías consultoras de software de la Web.

# Requisitos previos para PHP

La mayoría de los sitios que utilizan PHP también utilizan MySQL como base de datos. Si tiene pensado utilizar MySQL, debería bajar la fuente MySQL o la distribución binaria de [www.mysql.com/downloads/index.html](http://www.mysql.com/downloads/index.html). Lea la documentación de instalación para aprender a instalar MySQL en su sistema. También puede necesitar los paquetes de desarrollo MySQL.

Cubrir la instalación de MySQL para cada plataforma se escapa del objetivo de este libro.

En mi sistema Linux ejecutando Apache, he instalado:

- El servidor RPM para sistemas i386
- Programas clientes RPM para sistemas i386
- Archivos `Include` y librerías RPM para el desarrollo de sistemas i386
- Librerías RPM cliente compartidas para sistemas i386

Los archivos `include` y las librerías, son necesarios para PHP si quiere compilarlo con soporte MySQL.

## Compilar e instalar PHP

La versión actual de PHP es 4.0. Puede bajar código fuente PHP o la distribución de binarios de [www.php.net](http://www.php.net). En esta sección se supone que ha bajado la última distribución fuente de PHP, `php-4.1.0.tar.gz`.

**NOTA: Si se encuentra en una plataforma Windows, lea la ejecución de scripts PHP en el capítulo 21 para instalar PHP para Apache.**

Una vez que ha bajado la distribución fuente, extraiga la fuente en un directorio, utilizando el comando `tar xvzf php-4.1.0.tar.gz`. Recomiendo que lo instale en el mismo directorio en el que tenga instalada la fuente de Apache. Por ejemplo, si instaló la fuente de Apache en el directorio `/usr/local/src/httpd-2.0.16`, entonces extraiga el PHP en el directorio `/usr/local/src`. Se creará un nuevo subdirectorio llamado `php-4.1.0`.

En este momento, tiene que decidir cómo planear la ejecución de PHP. PHP puede ejecutarse como un módulo Apache (embebido en el servidor o como un módulo DSO) o como una solución CGI. La solución CGI significa que no tendrá ninguna ventaja de rendimiento con los scripts PHP sobre los scripts CGI normales, porque un intérprete PHP se cargará cada vez que se procese un script PHP en el modo CGI.

# Construir PHP como una solución CGI

Al igual que Perl, PHP puede utilizarse en forma de scripts independientes o embebido en páginas Web. Para construir el intérprete PHP para operaciones del modo CGI, haga lo siguiente:

1. Como raíz, cambie el directorio de la distribución fuente de PHP y ejecute:  
`./configure --enable-discard-path`
2. Ahora ejecute `make && make install` para compilar e instalar el intérprete PHP en su sistema.

## Construir PHP como un módulo Apache

Puede almacenar el módulo PHP dentro del binario de Apache o instalarlo como un módulo DSO para Apache. Una ventaja del módulo DSO es que se puede descargar simplemente comentándolo en una línea de configuración en `httpd.conf`, ahorrando, así, algo de memoria.

## Construir PHP como un módulo estático de Apache

Supongo que tiene instalada la fuente de Apache en el directorio `/usr/local/src/httpd-version` y la fuente PHP en el directorio `/usr/local/src/php-version`.

A continuación tiene el modo de compilar e instalar PHP como un módulo estático de Apache:

1. Si no tiene instalado Apache, ejecute el siguiente comando como raíz desde el directorio de la distribución fuente de Apache:  
`./configure --prefix=/usr/local/web/apache`

Puede añadir otras opciones si quiere. Ejecute `./configure --help` para determinar si hay otras opciones convenientes para sus necesidades.

2. Ahora ejecute la siguiente línea desde el directorio de la distribución fuente de PHP:

```
./configure --with-apache=../httpd-version \
--with-mysql=/usr
```

Aquí la opción `--with-mysql` está asignada a `/usr` porque los paquetes RPM de MySQL instalan los archivos `include` en el directorio `/usr/include/mysql`. Si su sistema tiene `includes` de MySQL en una localización distinta, debería utilizar un nombre distinto para el directorio. Puede descubrir dónde se guardan los `includes` MySQL utilizando el comando `locate mysql.h`, que está disponible en la mayoría de los sistemas Unix con la característica `locate` de las bases de datos.

**NOTA:** No necesita la opción `--with-mysql` si no va a utilizar MySQL con PHP, o si quiere utilizar soporte integrado PHP para MySQL, que sólo se recomienda si no va a utilizar MySQL como base de datos para Apache.

3. Ejecute el comando `cp php.ini-dist /usr/local/lib/php.ini` para copiar el archivo `php.ini-dist` en `/usr/local/lib` como `php.ini`. Éste es el archivo de configuración PHP que se discutirá más tarde en la sección "Configurar PHP utilizando `php.ini`".
4. Ejecute `make && make install` para compilar e instalar la parte PHP del código fuente.
5. A continuación cambie el directorio de la distribución fuente de Apache y ejecute:

```
/configure --prefix=/usr/local/apache\  
          --activate-module= modules/php4/libphp4.a
```

Puede añadir cualquier otra opción en la línea de comandos anterior. Ejecute el comando `./configure --help` para obtener información sobre otras opciones.

6. Ejecute `make && make install` para compilar e instalar Apache con soporte PHP.
7. Ejecute el comando `/usr/local/apache/bin/apachectl restart` para reiniciar (o iniciar) Apache.

## Construir PHP como un módulo Dynamic Shared Object (DSO)

Debe tener activado el soporte DSO en Apache antes de utilizar PHP como un módulo DSO. Para volver a compilar Apache con soporte DSO, siga los siguientes pasos:

1. Desde el directorio de la distribución fuente de Apache ejecute el siguiente comando como raíz:

```
./configure --prefix=/usr/local/apache --enable-so
```

Puede añadir también otras opciones si lo considera necesario.

2. Puede compilar e instalar Apache utilizando el comando `make && make install`.

Una vez que tiene el servidor Apache con el soporte DSO activado, puede hacer lo que mostramos a continuación para crear un módulo DSO para PHP:

1. Desde el directorio de la distribución fuente de PHP ejecute el siguiente comando como raíz:

```
./configure --with-apxs2=/usr/local/apache/bin/apxs \
--with-mysql=/usr
```

Aquí, la opción `--with-mysql` está asignada a `/usr` porque los paquetes RPM de MySQL instalan los archivos `include` en el directorio `/usr/include/mysql`. Si su sistema tiene inclusiones (archivos `include`) de MySQL en una localización distinta, debería utilizar un nombre distinto para el directorio. Puede descubrir dónde se guardan los `includes` MySQL utilizando el comando `locate mysql.h`, que está disponible en la mayoría de los sistemas Unix con la característica `locate` de las bases de datos.

2. Ahora ejecute `make && make install` para compilar e instalar la versión DSO del módulo PHP para Apache.
3. Ejecute el comando `/usr/local/apache/bin/apachectl` para reiniciar (o iniciar) Apache.

## Configurar Apache para PHP

Una vez que tiene instalado el módulo `mod_php` para Apache y configurado `php.ini` tal y como se discutió antes, está preparado para configurar Apache para PHP del siguiente modo:

1. Si tiene compilado e instalado PHP como un módulo DSO para Apache, añada la siguiente línea al archivo `httpd.conf`:  
`LoadModule php4_module modules/libphp4.so`
2. Añada las líneas siguientes al archivo `httpd.conf`

```
<Files *.php>
    SetOutputFilter PHP
    SetInputFilter PHP
</Files>
```

Esto le indica a Apache que cualquier archivo con la extensión `.php` debe tratarse como un script PHP y procesarse utilizando los filtros de entrada y salida de PHP.

**TRUCO:** No hay razón para utilizar una extensión diferente para los scripts PHP. Por ejemplo, puede asignar la directiva `AddType` a `AddType application/x-httdp-php .html` y tener todas sus páginas HTML tratadas como un script PHP. No recomiendo utilizar la extensión `.html` porque hay posibilidad de que muchas de sus páginas HTML no sean scripts PHP, y no querrá que el servidor Web se haga más lento al tener que analizar cada página buscando scripts PHP.

3. Guarde el archivo `httpd.conf` y reinicie el servidor Web Apache como siempre.

Ahora está listo para crear scripts PHP para su sitio Web. Puede crear scripts PHP y almacenarlos en cualquier sitio del árbol de documentos de su sitio Web y Apache los procesará automáticamente como scripts PHP.

## Configurar PHP utilizando `php.ini`

El archivo de configuración de PHP se llama `php.ini`, y está almacenado en el directorio `/usr/local/lib`. Cuando se carga un módulo PHP lee el archivo `php.ini`. El módulo busca el archivo `php.ini` en el directorio que está funcionando, en la ruta designada por la variable de entorno `PHPINC`, y en `/usr/local/lib`.

**NOTA:** Si está utilizando PHP como una solución CGI, `php.ini` se lee cada vez que se ejecuta un CGI PHP. Por otro lado, cuando se carga PHP como un módulo Apache, se lee una sola vez. Debe reiniciar el servidor Apache utilizando el comando `/usr/local/apache/bin/apachectl` para volver a cargar cualquier cambio que realice en el archivo `php.ini`.

## Directivas PHP en `httpd.conf`

Con la versión PHP 4 sólo están permitidas cuatro directivas específicas de `mod_php` en `httpd.conf`, tal y como se muestra en las siguientes secciones. El resto de las directivas PHP deben estar en el archivo `php.ini`.

### `php_admin_flag`

La directiva `php_flag` le permite asignar un valor boleando (On o Off) para un parámetro de configuración. Esta directiva no puede aparecer en contenedores de directorios ni en archivos `.htaccess`.

**Sintaxis:** `php_flag name On | Off`

**Contexto:** configuración del servidor, host virtual

### `php_admin_value`

La directiva `php_admin_value` le permite asignar un valor para un parámetro de configuración. Esta directiva no puede aparecer en contenedores de directorios ni en archivos `.htaccess`.

**Sintaxis:** `php_admin_value name value`

**Contexto:** configuración del servidor, host virtual

## **php\_flag**

La directiva `php_flag` le permite asignar un valor boleando (On o Off) a un parámetro de configuración.

**Sintaxis:** `php_flag name On | Off`

**Contexto:** configuración del servidor, host virtual, directorio, ámbito de directorio (.htaccess)

Por ejemplo:

```
php_flag display_errors On
```

## **php\_value**

La directiva `php_value` le permite asignar un valor a un parámetro de configuración.

**Sintaxis:** `php_value name value`

**Contexto:** configuración del servidor, host virtual, directorio, ámbito de directorio (.htaccess)

Por ejemplo:

```
php_value error_reporting 15
```

# **Directivas PHP en php.ini**

El archivo `php.ini` tiene una sintaxis sencilla con una estructura directiva = valor. Se ignoran las líneas que consisten en un punto y coma inicial, y las líneas con espacios en blanco.

Los nombres de secciones van encerrados entre corchetes. Puede ver estas directivas de `php.ini` en [www.php.net/manual/en/configuration.php](http://www.php.net/manual/en/configuration.php). Las siguientes secciones discuten las más útiles.

## **auto\_append\_file**

La directiva `auto_append_file` le permite asignar un pie de página de documento a cada página analizada por PHP.

**Sintaxis:** `auto_append_file filename`

## **auto\_prepend\_file**

La directiva `auto_prepend_file` le permite asignar una cabecera de documento a cada página analizada por PHP.

**Sintaxis:** `auto_prepend_file filename`

En el siguiente ejemplo la página preload.php se cargará antes de que se procese cada página PHP. Esta página es un buen sitio para establecer conexiones con bases de datos.

```
auto-prepend-file preload.php
```

## **default\_charset**

La directiva default\_charset determina el conjunto de caracteres por defecto.

**Sintaxis:** default\_charset char\_set

El siguiente ejemplo determina que el conjunto de caracteres por defecto es 8-bit UTF:

```
default_charset = "UTF-8"
```

## **disable\_functions**

La directiva disable\_functions le permite desactivar una o más funciones por razones de seguridad.

**Sintaxis:** disable\_functions function\_name [function\_name]

Puede especificar una lista, delimitada por comas, de funciones PHP, tal y como se muestra a continuación:

```
disable_functions = fopen, fwrite, popen
```

En este caso, la función responsable de abrir y escribir archivos o pipes (tuberías) está desactivada.

**NOTA:** Esta directiva no se ve reflejada en el resultado de la ejecución de PHP.

## **display\_errors**

La directiva display\_errors activa o desactiva la impresión de un mensaje de error en la pantalla. Se recomienda únicamente utilizarla en sistemas de desarrollo y no para los servidores de producción. Para los sistemas de producción, debe utilizar log\_errors junto con error\_log para registrar mensajes de error a archivos o a un servidor syslog.

**Sintaxis:** display\_errors On | Off

## **enable\_dl**

La directiva enable\_dl activa o desactiva la capacidad de cargar dinámicamente una extensión PHP.

**Sintaxis:** enable\_dl On | Off

**Predefinido:** enable\_dl On

## **error\_append\_string**

La directiva `error_append_string` asigna el string que se adjunta al mensaje de error. Ver la directiva `error_prepend_string`.

**Sintaxis:** error\_append\_string string

## **error\_log**

La directiva `error_log` asigna la ruta de registro de error PHP. Puede especificar un nombre de ruta completo del archivo de registro, o puede especificar la palabra clave `syslog` en sistemas Unix para registrar utilizando la facilidad `syslog`. En los sistemas Windows, asignar esta directiva a `syslog` escribe entradas de registro en el registro de eventos de Windows.

**Sintaxis:** error\_log fqpn

## **error-prepend\_string**

La directiva `error-prepend_string` asigna el string que está adjunto a un mensaje de error. Esta directiva se utiliza con `error_append_string`.

**Sintaxis:** error-prepend\_string string

Los mensajes de error se muestran en la pantalla. Por ejemplo, utilizando estas dos directivas, puede imprimir mensajes de error en color rojo:

```
error-prepend_string = "<font color=red>"  
error_append_string = "</font>"
```

## **error\_reporting**

La directiva `error_reporting` le permite especificar un campo de bit para especificar un nivel de informe de errores. El campo de bit se puede construir utilizando las constantes predefinidas mostradas en la tabla 15.1.

**Sintaxis:** error\_reporting [bit field] [predefined\_constant]

**Tabla 15.1.** Constantes para la directiva `error_reporting`

Constante	Significado
E_ALL	Muestra todos los errores, advertencias y noticias.
E_ERROR	Muestra sólo los errores fatales en tiempo de ejecución.
E_WARNING	Muestra las advertencias en tiempo de ejecución.

Constante	Significado
E_PARSE	Muestra errores de análisis.
E_NOTICE	Muestra noticias de problemas posiblemente en el código.
E_CORE_ERROR	Muestra errores fatales que tienen lugar durante el arranque inicial de PHP.
E_CORE_WARNING	Muestra advertencias que tienen lugar durante el arranque inicial de PHP.
E_COMPILE_ERROR	Muestra errores fatales en tiempo de compilación.
E_COMPILE_WARNING	Muestra advertencias en tiempo de compilación (errores no fatales).
E_USER_ERROR	Mensaje de error generado por el usuario.
E_USER_WARNING	Mensaje de advertencia generado por el usuario.
E_USER_NOTICE	Mensaje de noticias generado por el usuario.

Puede utilizar operadores de tratamientos de bits como ~ (invierte), & (operador de tratamientos de bits que significa "AND"), y | (operador de tratamientos de bits que significa "OR") para crear un nivel de informe de errores personalizado. Por ejemplo:

```
Error_reporting = E_ALL & ~E_WARNING & ~E_NOTICE
```

Esto le dice a la ingeniería PHP que muestre todos los errores excepto las advertencias y las noticias. No se recomienda desplegar mensajes de error en un servidor de producción. Debería desplegar errores únicamente en su sistema de desarrollo o durante la fase de desarrollo del servidor de producción. En un servidor de producción, utilice las directivas `log_errors` y `error_log` para escribir registros en los archivos o en la utilidad `syslog`.

## extension

La directiva `extension` le permite cargar un módulo de extensión para el propio PHP.

**Sintaxis:** `extension module_name`

Por ejemplo, la siguiente directiva carga la extensión GD de la librería de gráficos para PHP en Windows mediante la directiva `extension`. La ingeniería PHP carga este tipo de módulos dinámicos en el arranque del servidor.

```
extension=php_gd.dll
```

De forma parecida, la siguiente directiva carga el módulo MySQL DSO para Apache en la plataforma Unix con soporte DSO:

```
extension=mysql.so
```

Puede repetir `extension` tantas veces como necesite cargar módulos distintos.

## **extension\_dir**

La directiva `extension_dir` define el directorio en el que se almacenan los módulos PHP que se cargan dinámicamente. El valor por defecto es adecuado para la mayoría de las instalaciones PHP.

**Sintaxis:** `extension_dir directory`

**Predefinido:** `extension_dir ./`

## **implicit\_flush**

La directiva `implicit_flush` activa o desactiva la descarga implícita de la salida del script al igual que utiliza `print`, `echo` y los bloques HTML para las salidas. Cuando está activada, esta directiva emitirá una llamada `flush()` después de cada salida `print()`, `echo` o de un bloque HTML. Esto resulta verdaderamente útil para propósitos de depuración, pero también una disminución en el rendimiento en el entorno de producción. Sólo se recomienda para sistemas de desarrollo.

**Sintaxis:** `implicit_flush On | Off`

**Predefinido:** `implicit_flush Off`

## **include\_path**

La directiva `include_path` asigna la ruta para las funciones `include()` y `require()`. Puede realizar una lista con varios directorios.

**Unix:** `include_path path[:path]`

**Sintaxis Windows:** `include_path path[;path]`

Por ejemplo, las siguientes líneas determinan que PHP debe buscar los archivos `include` primero en el directorio `/usr/local/lib/php` y después en el directorio actual:

```
include_path = /usr/local/lib/php:.
```

En Windows, esta directiva se asigna del siguiente modo:

```
include_path = /usr/local/lib/php;.
```

## **log\_errors**

La directiva `log_errors` activa o desactiva el registro de errores PHP. Debe utilizar la directiva `error_log` para especificar una ruta de registro o un archivo `syslog`.

**Sintaxis:** `log_errors On | Off`

## **magic\_quotes\_gpc**

La directiva `magic_quotes_gpc` activa o desactiva las comillas de escape (los caracteres comillas, dobles comillas, nulo y la barra inversa) para los datos GET, POST y cookie.

**Sintaxis:** `magic_quotes_gpc On | Off`

**Predefinido:** `magic_quotes_gpc On`

## **magic\_quotes\_runtime**

La directiva `magic_quotes_runtime` activa o desactiva las comillas de los textos generados internamente. En otras palabras, si recupera un registro de una base de datos que tiene una etiqueta del tipo `<?php anything goes here ?>` embebida en él, no se procesará y no se tratará como código PHP el contenido que se encuentra en el interior de las etiquetas (que forma parte de los datos).

**Sintaxis:** `magic_quotes_runtime On | Off`

**Predefinido:** `magic_quotes_runtime = Off`

## **max\_execution\_time**

La directiva `max_execution_time` determina el tiempo máximo que se puede ejecutar un script para producir una salida. Una vez que el script ha tomado un tiempo determinado, PHP finaliza el script. A no ser que tenga pensado ejecutar scripts PHP que lleven mucho tiempo, el valor por defecto debería ser suficiente para la mayor parte de las situaciones.

**Sintaxis:** `max_execution_time seconds`

**Predefinido:** `max_execution_time 30`

## **memory\_limit**

La directiva `memory_limit` determina la cantidad máxima de RAM que un script PHP puede consumir. El valor por defecto, 8MB, debería ser suficiente para scripts PHP pequeños o medianos. También puede especificarlo en bytes.

**Sintaxis:** `memory_limit bytes [nM]`

**Predefinido:** `memory_limit 8M`

Por ejemplo, las dos líneas siguientes son equivalentes entre sí:

```
memory_limit 8M  
memory_limit 8388608
```

## **output\_buffering**

La directiva `output_buffering` le permite activar o desactivar el buffering (selección de datos para editarlos o procesarlos antes de llevarlos a un archivo o bases de datos) de salidas. Cuando tiene el valor `On`, puede imprimir cabeceras HTTP en cualquier parte de un script PHP. Ser capaz de sacar una cabecera en la mitad de un script, incluso tras imprimir otro contenido, significa que un script puede mostrar una página de error incluso si tuvo cierto éxito anteriormente.

**Sintaxis:** `output_buffering On | Off`

**Predefinido:** `output_buffering On`

Además, puede utilizar las directivas integradas `ob_start()` y `ob_end_flush()` para iniciar y finalizar la descarga de contenido directamente. Por ejemplo:

```
<?php  
  
ob_start(); // Buffer toda la salida  
  
echo "Buffered contents \n";  
  
ob_end_flush(); // Saca la página, descarga el buffer de la  
salida.  
  
?>
```

La salida ha pasado por un proceso buffer.

## **safe\_mode**

La directiva `safe_mode` determina el modo de seguridad para PHP cuando se utiliza como una solución CGI. Cuando tiene el valor `On`, esta directiva asegura que los scripts PHP ejecutados por el intérprete PHP en el modo CGI, no permiten ningún acceso más allá del directorio de documentos del sitio Web.

**Sintaxis:** `safe_mode On | Off`

**Predefinido:** `safe_mode Off`

## **safe\_mode\_allowed\_env\_vars**

La directiva `safe_mode_allowed_env_vars` le permite determinar un prefijo para todas las variables de entorno que un usuario puede cambiar utilizando

do la función `putenv()`. El valor por defecto le permite a los usuarios cambiar cualquier variable de entorno que comience con el prefijo PHP.

**Sintaxis:** `safe_mode_allowed_env_vars prefix`

**Predefinido:** `safe_mode_allowed_env_vars PHP_`

### **safe\_mode\_protected\_env\_vars**

La variable `safe_mode_protected_env_vars` le permite determinar una lista, delimitada por comas, de variables de entorno que no se pueden cambiar por ningún script PHP que utilice la función `putenv()`.

**Sintaxis:** `safe_mode_protected_env_vars environment_variable [environment_variable ...]`

**Predefinido:** `safe_mode_protected_env_vars = LD_LIBRARY_PATH`

Si quiere proteger todas las variables de entorno que empiecen con el prefijo `HTTP_`, puede utilizar:

```
safe_mode_protected_env_vars = HTTP_
```

### **track\_errors**

La directiva `track_errors` activa o desactiva el almacenaje de mensajes de error en un variable PHP llamada `$php_errormsg`.

**Sintaxis:** `track_errors On | Off`

### **upload\_max\_filesize**

La directiva `upload_max_filesize` determina el tamaño máximo de un archivo que se puede descargar mediante PHP. El límite por defecto es 2MB (2M). También puede especificar el número en kilobytes.

**Sintaxis:** `upload_max_filesize kilobytes`

**Predefinido:** `upload_max_filesize 2M`

Por ejemplo, las dos líneas siguientes son equivalentes:

```
upload_max_filesize = 2M  
upload_max_filesize = 2097152
```

### **upload\_tmp\_dir**

La directiva `load_tmp_dir` define la localización temporal del directorio para los archivos descargados mediante PHP: es habitual asignar el valor `/tmp` en los sistemas Unix; en los sistemas Windows, lo normal es asignar `/temp` o sin dejarlo sin asignar, en cuyo caso, PHP utiliza el valor por defecto.

**Sintaxis:** `load_tmp_dir directory`

# Trabajar con PHP

Esta sección le muestra unos cuantos ejemplos de cómo puede utilizar scripts PHP para generar contenido Web dinámico.

## Crear un script PHP sencillo desde la línea de comandos

Una vez que tiene configurado PHP, puede ejecutar un script PHP desde la línea de comandos del mismo modo que hacemos con un script de Perl. Por defecto, el intérprete PHP está instalado en el directorio en el que lo extrajo desde la distribución. Puede crear un script PHP sencillo llamado `test.php` como el que se muestra en el listado 15.1.

**Listado 15.1.** `test.php`

```
#!/usr/local/bin/php -q
# Para Windows cambia la linea anterior a:
#!Drive:/path/to/php -q
#
#
<?php
echo "Welcome to the PHP World\n";
?>
```

Puede ejecutar este script desde la línea de comandos utilizando `/usr/local/bin/php test.php`, o puede cambiar el permiso de archivo utilizando `chmod +x test.php` y ejecutándolo utilizando `./test.php` del directorio del script. La opción `-q` que se utiliza aquí le dice a PHP que suprima el content-type de las cabeceras HTTP, que se imprime para hacer compatible la salida Web.

## Crear páginas Web PHP

Crear un script PHP para su sitio Web es tan sencillo como el ejemplo de la última sección. El listado 15.2 muestra un sencillo script llamado `hello.php`.

**Listado 15.2.** `hello.php`

```
<html>
  <head><title>Hello World Script</title>
<body>
```

```
<?php echo "Hello World from PHP"; ?>  
</body>  
</html>
```

Cuando guarda el script en el directorio dentro del árbol de documentos de su servidor Web, puede acceder a él mediante `http://your_web_server/path/to/hello.php`. Si ha utilizado una directiva como `AddType application/x-httdp-php.php` tal y como se recomendaba en la sección "Configurar Apache para PHP", entonces el módulo PHP analizará y ejecutará este script PHP de una sola línea, que imprime la sentencia "Hello World from PHP". Observe que PHP imprime automáticamente la cabecera `Content-Type` apropiada (`text/html`).

A continuación vamos a ver un script algo más útil, el que se muestra en el listado 15.3, que muestra información sobre cómo está instalado PHP en su servidor Web.

#### Listado 15.3. info.php

```
<html>  
    <head><title>Hello World Script</title>  
<body>  
  
<?php  
    phpinfo();  
?>  
  
</body>  
</html>
```

Esta página llama a una función PHP llamada `phpinfo()` para que muestre una gran cantidad de opciones PHP tal y como se han configurado utilizando `php.ini`.

## Utilizar un script PHP como un Server-Side Include

Si quiere utilizar scripts PHP en llamadas Server-Side Include (SSI), puede llamar a su script PHP utilizando la siguiente etiqueta SSI:

```
<!--#include virtual="/path/script_name.php"-->
```

Por ejemplo:

```
<!--#include virtual="/phpssi/test.php"-->
```

Aquí el script PHP llamado `/phpssi/test.php` se cargará desde la página que utiliza las llamadas SSI anteriores. Para que funcionen los scripts PHP con SSI, debe hacer lo siguiente:

1. Activar la opción ExecCGI en el directorio que contiene los scripts PHP que hay que ejecutar mediante las llamadas SSI.
2. Asegurarse de que la opción IncludesNoExec está desactivada para ese mismo directorio. Por ejemplo:

```
DocumentRoot "/www/mysite/htdocs"

<Directory "/www/mysite/htdocs/parsed">
    Options +Includes

    AddType text/html .shtml

    <FilesMatch "\.shtml[.]$">
        SetOutputFilter INCLUDES
    </FilesMatch>

</Directory>

<Directory "/www/mysite/htdocs/php">
    Options +ExecCGI

    <Files *.php>
        SetOutputFilter PHP
        SetInputFilter PHP
    </Files>

</Directory>
```

En el segmento de configuración anterior, todos los archivos que acaban con la extensión .shtml del directorio /www/mysite/htdocs/parsed se tratan como páginas SSI y los archivos de /www/mysite/htdocs/php se tratan como scripts PHP que, además, se pueden ejecutar mediante llamadas SSI. Ahora, una página SSI en el directorio /www/mysite/htdocs/parsed puede llamar a un script de /www/mysite/htdocs/php mediante la siguiente llamada SSI:

```
<!--#include virtual="/php/script_name.php"-->
```

## Utilizar una página PHP para un directorio index

Si tiene activado el soporte PHP en httpd.conf tal y como se discutió en la sección "Configurar Apache para PHP," también puede utilizar scripts PHP como una página de index de un directorio. Por ejemplo:

```
DirectoryIndex index.html index.php
```

Aquí la directiva asigna index.html como la página index preferida para un directorio, pero si index.html no existe, se utiliza index.php.

Sin embargo, si quiere utilizar scripts PHP como la página index por defecto, entonces tiene que invertir el orden de los nombres de archivo para que la página index.php se encuentre colocada antes que el resto. Por ejemplo:

```
DirectoryIndex index.php index.html
```

Ahora Apache buscará primero el script index.php.

## Utilizar archivos include

PHP le permite incluir un archivo externo de librerías con código PHP o un archivo HTML corriente en un script. Por ejemplo:

```
<html>
<head>
    <title> PHP Resource Site </title>
</head>

<body>
<h1>PHP Resource</h1>
<hr>

<p>Welcome to PHP Resource Site. </p>

<?php
    include('/www/mysite/htdocs/global/copyright.html');

?>

</body>
</html>
```

Aquí, el script PHP es una sola llamada a la función include('/www/mysite/htdocs/global/copyright.html');, que le dice a PHP que incluya un archivo llamado /www/mysite/htdocs/global/copyright.html. Esta página copyright.html está incluida en la página actual, al igual que la etiqueta SSI <!--#include virtual="/path/filename"-->. Por otro lado, el archivo incluido puede ser un script PHP. Por ejemplo:

```
include('/www/mysite/htdocs/global/copyright.php');
```

carga un script PHP llamado copyright.php desde dicho directorio. El script del archivo incluido será ejecutado. Cualquier código del archivo incluido tiene acceso a las variables del script padre. Por ejemplo, el script copyright.php tiene acceso a la variable \$year.

```
<?php
$year = '2001';
```

```
include('/www/mysite/htdocs/global/copyright.php');  
?>
```

Por eso, el script `copyright.php` puede contener una sentencia como `echo "The current year is: $year \n";`. Puede incluir varios archivos, o puede incluir el mismo archivo varias veces. El script `include` se ejecutará cada vez. Por ejemplo:

```
<?php  
  
$year = '2001';  
  
include('/www/mysite/htdocs/global/copyright.php');  
  
$year = '2002'  
  
include('/www/mysite/htdocs/global/copyright.php');  
  
?>
```

Si el script `copyright.php` consiste en un script PHP sencillo como `<?php echo "The current year is: $year\n"; ?>`, entonces el script anterior mostrará:

```
The current year is: 2001  
The current year is: 2002
```

También puede utilizar `include_once()` para incluir un script PHP o un archivo HTML estático. Sin embargo, si utiliza `include_once()` en lugar de `include()`, el código del script incluido se ejecutará una sola vez. Por ejemplo:

```
<?php  
  
$year = '2001';  
  
include_once('/www/mysite/htdocs/global/copyright.php');  
  
$year = '2002'  
  
include_once('/www/mysite/htdocs/global/copyright.php');  
  
?>
```

Aquí, la salida para `copyright.php` es:

```
The current year is: 2001
```

El segundo `include_once()` no ejecutará de nuevo el `copyright.php`. Puede utilizar las funciones `include()` o `include_once()` para centralizar código común o archivos HTML necesarios para varias páginas PHP. También

puede utilizar las funciones `require()` y `require_once()`. La función `require()` se reemplaza a sí misma con el contenido del archivo nombrado. Este reemplazo tiene lugar durante la compilación del código que realiza la ingeniería PHP, no durante la ejecución de la llamada al script. La función `require_once()` únicamente inserta el contenido del script requerido una sola vez.

**NOTA:** Las funciones `include()`, `include_once()`, `require()` y `require_once()`, pueden acceder a cualquier archivo del sistema de archivos siempre que el ID del usuario, que está utilizando para ejecutar el servidor Web, tenga acceso de lectura al archivo.

## Mejorar el manejo de errores con PHP

Apache proporciona una directiva llamada `ErrorDocument` que le permite mostrar una página de error personalizada para un error del servidor. Por ejemplo:

```
ErrorDocument 404 /missing.html
```

Cada vez que encuentre una solicitud de un archivo o un directorio inexistente, envía una respuesta `Status: 404 Not Found` response y entonces muestra la página `/missing.html`. Puede cambiar esta directiva:

```
ErrorDocument 404 /missing.php
```

Lo que da como resultado a la ejecución del script PHP llamado `missing.php` cada vez que se encuentre el mismo error. Sin embargo, Microsoft IE detecta una respuesta `40x` y muestra un mensaje de error personalizado que genera él. Para evitar este tipo de páginas, puede hacer que la página `missing.php` devuelva una respuesta `Status: 200 OK`. Por ejemplo con el código siguiente:

```
<?php  
  
header("Status: 200 OK\n");  
header("Location: /missing.html");  
  
?>
```

el script `missing.php` muestra la cabecera HTTP e instruye a Apache para que redirija al cliente Web a `/missing.html`.

## Procesar formularios Web con PHP

PHP convierte el procesamiento de formularios en algo sencillo. Puede acceder a los campos de datos del formulario Web como si fueran variables dentro de

un script PHP. El listado 15.4 muestra un formulario Web sencillo que tiene un solo campo de datos llamado `keywords`.

**Listado 15.4.** simple\_form.html

```
<html>
<head><title>Simple HTML Form</title>
</head>

<body>

<form action="form.php">
<input type="text" name="keywords" size=30>
<input type=submit>

</form>
</body>
</html>
```

Cuando un usuario envía este formulario mediante un navegador Web, el navegador envía una solicitud GET del tipo `http://server/form.php?keywords=value_entered_in_the_form`.

Por ejemplo, si introduce la palabra clave `makesite` y envía el formulario a `www.domain.com`, el servidor Web recibirá una solicitud para `http://www.domain.com/form.php?keywords=makesite`.

El script `form.php` responsable de procesar este formulario Web se muestra en el listado 15.5.

**Listado 15.5.** form.php

```
<?php
echo "You have entered $keywords keyword(s).\n";
?>
```

Este sencillo script PHP imprime las palabras claves introducidas en el campo `keywords` del formulario HTML llamado `form.html`.

Para la solicitud siguiente:

`http://www.domain.com/form.php?keywords=makesite`

el script imprime:

`You have entered makesite as keyword(s).`

Aquí la variable `$keywords` tiene asignado el valor `makesite'` porque `keywords=makesite` es la cadena de consulta que se pasa al servidor mediante el método GET. Si la línea `<form action="form.php">` del formulario Web, se cambia a `<form action="form.php" method="POST">`,

entonces se utilizará el método POST HTTP y el módulo PHP leerá el dato keywords=makesite y automáticamente se hará disponible al script como el valor de la variable \$keywords del script.

Si el formulario Web tiene varios campos, todos los campos se convierten en variables de script que se nombran de forma acorde. Por ejemplo, en el listado 15.6 se muestra una versión modificada de simple\_form.html.

#### Listado 15.6. modified\_simple\_form.html

```
<html>
<head>
    <title>Simple HTML Form</title>
</head>

<body>
<form action="modified_form.php" method="POST">

<input type="text" name="keywords" size=30>
<select name="case">
<option value="lower">Lower case</option>
<option value="upper">Upper case</option>
<option value="dontcare">Don't Care</option>
</select>

<input type=submit>

</form>
</body>
</html>
```

Aquí, el formulario tiene dos campos de entrada de datos (keywords y case).

Cuando este formulario Web se envía mediante el método POST, modified\_form.php (mostrado en el listado 15.7) recibe todos los datos.

#### Listado 15.7. modified\_form.php

```
<?php
print "You entered:<br>Keywords: $keywords <br>Case: $case\n";
?>
```

Por ejemplo, cuando un usuario introduce makesite en el campo keywords y elige la opción dontcare (no le importa el tipo de letra) de la lista desplegable, el script (modified\_form.php) muestra lo siguiente:

```
You entered:
Keywords: makesite
Case: upper
```

# Crear sesiones con PHP

PHP 4 introduce gestión nativa de sesiones, que facilita al usuario las sesiones de creación. Esto significa que puede seguir la pista de un usuario mientras que el usuario se mueve de una página a otra. Hay dos modos de mantener sesiones con PHP.

## Utilizar cookies HTTP para crear sesiones de usuario

Este es el método por defecto. Para crear una sesión tiene que llamar a la función `session_start()` en su página PHP. Como ejemplo tenemos la página PHP llamada `start.php`, que se muestra en el listado 15.8.

Listado 15.8. `start.php`

```
<?php
    session_start();
    session_register('count');
    $count++;
?>
<html>
<head><title>Start Session</title>
</head>
<body bgcolor="Red">

<?php echo "You have visited this page $count time(s)."; ?>

<p>
<a href="/next_page.php">Next</a>
</body>
</html>
```

En el listado puede ver dos scripts PHP. El primer script crea una sesión llamando a la función `session_start()`. Además registra una variable de sesión llamada `count` utilizando la función `session_register()` e incrementa la variable en uno. Observe que este script se mantiene fuera de la etiqueta `<html>`. Esto es necesario porque el script asigna la cookie de sesión, que es una cabecera HTTP que debe aparecer antes que cualquier contenido. La página PHP anterior, supone que tiene `output_buffering` con el valor `Off` en `php.ini`. Esta es la asignación por defecto porque el buffering de la salida da lugar a que las páginas PHP sean más lentas. Si activa el buffering de las salidas asignando `output_buffering = On` en `php.ini`, puede situar los scripts PHP en cualquier posición de la página.

El segundo script simplemente imprime el valor de `$count`. A continuación, si carga esta página por primera vez, utilizando un navegador Web, la función `session_start()` creará una nueva sesión basada en una cookie HTTP. La cookie enviada a su navegador Web tendrá atributos parecidos a los siguientes:

```
Cookie Name: PHPSESSID  
Domain : 192.168.1.100  
Path : /  
Expires : End of session  
Secure : No  
Data : 3de4aa1f73e33dd8f2c8b8d9f69e442e
```

El nombre de la cookie que se envía al navegador Web es PHPSESSID. Este es el nombre por defecto asignado en `php.ini` utilizando la línea siguiente:

```
session.name = PHPSESSID
```

Puede cambiarlo a otro nombre si quiere. Por defecto, la cookie es válida para únicamente el servidor Web en sí. El ejemplo de la cookie anterior muestra que el campo de dominio de la cookie está asignado a la dirección IP del servidor Web. Si quiere que una cookie de sesión sea válida a lo largo de varios servidores Web dentro de su dominio, puede asignarle a la directiva `session.cookie_domain` en el archivo `php.ini`, el nombre del dominio de máximo nivel de sus sitios Web.

Por ejemplo, si tiene tres sitios Web, `corp.domain.com`, `www.domain.com` y `extranet.domain.com`, y le gustaría que la cookie de sesión fuese válida para los tres sitios, puede asignarle el siguiente valor a la directiva `session.cookie_domain`:

```
session.cookie_domain = .domain.com
```

Observe el punto delante del nombre de dominio de máximo nivel (`domain.com`); este punto le permite a la cookie ser válida en todos los host que se encuentren bajo este dominio.

La cookie de sesión por defecto, expira al final de una sesión. En otras palabras, cuando se cierra un navegador Web, la cookie de sesión expira. Este comportamiento por defecto se fija con la directiva `session.cookie_lifetime` en el archivo `php.ini`. Por defecto tiene el valor 0, lo que da lugar a que la cookie expire al final de una sesión (es decir, cuando el usuario cierra el navegador):

```
session.cookie_lifetime = 0
```

**NOTA: PHP no almacena datos del usuario dentro de la cookie. La cookie es simplemente un identificador que señala al dato del usuario en el lado del servidor.**

Cuando el primer script del listado 15.8 crea la sesión, puede acceder fácilmente a la variable o a las variables de sesión de otras páginas. Por ejemplo. El listado 15.9 muestra `next_page.php`, que utiliza la variable llamada `$count` creada en `start.php`.

### Listado 15.9. next\_page.php

```
<html>
<head><title>Using Session Variables</title>
</head>
<body bgcolor="Red">

<?php
    session_start();
    $count++;
    echo "You have visited this page $count time(s).";
?>

</body>
</html>
```

Cada vez que quiera utilizar una sesión actual en una página, debe utilizar el método `session_start()`. Esto resulta algo complicado porque podría dar lugar a que piense que está creando una sesión nueva. En realidad, `session_start()` detecta la sesión existente y la utiliza. Las variables de sesión se vuelven disponibles una vez que ha realizado la llamada a la función `session_start()`. Observe que en esta página, no situamos el script antes del contenido. En otras palabras, no situamos el script antes de la etiqueta `<html>` porque no vamos a seguir editando una nueva sesión, por lo que no tenemos que fijar una nueva cookie. Sin embargo, esto supone que tiene ya una sesión. Para ser capaz de crear una sesión nueva sin que ya exista una, mueva el script fuera del contenido como en la página `start.php` o active `output_buffering` en `php.ini`. Además, si ha pensado permitir la creación de una sesión en cualquier página, tiene que utilizar el método `session_register()` para registrar variables tal y como se muestra en `start.php`.

En cualquier página en la que quiera utilizar `session_start()` para seleccionar la sesión actual, puede utilizar también `session_register()` para añadir nuevas variables a la sesión de usuario.

### Utilizar codificación de URL para crear sesiones de usuario

Si no quiere depender de las cookies para mantener una sesión, puede utilizar sesiones codificadas por URL. Asignándole a la directiva `session.use_cookies` el valor 0 en `php.ini` activamos esta posibilidad. Sin embargo, a diferencia de lo que ocurría con las cookies, en este caso necesita una ID de sesión (asignado por `session.name` en `php.ini`) utilizando `<?=SID?>` como parte de las URL en una página. Por ejemplo, el listado 15.10 muestra una página llamada `start_url_encoded_session.php`.

### Listado 15.10. start\_url\_encoded\_session.php

```
<?php
    session_start();
```

```

    session_register('count');
    $count++;
?>

<html>
<head><title>Session Management</title>
</head>
<body bgcolor="Red">

<p>
<?php echo "You have visited this page $count time(s)."; ?>

<p>
<a href="/next_page.php?<?=SID?>">Next Page</a>
</body>
</html>

```

Observe el enlace Next Page. Cuando un usuario hace clic en este enlace, el navegador solicita /next\_page.php y se pasa una cadena de consulta llama da PHPSESSID=session\_identifier al servidor Web. Por ejemplo:

```

http://207.183.233.21/
next_page.php?PHPSESSID=6b2cee31528316080ff88fe81f800bf8

```

Esto permite que el script PHP (next\_page.php) sepa a qué sesión pertenece el usuario y que evite la utilización de cookies HTTP. Si quiere mantener la sesión utilizando este método, todos los enlaces de página PHP tienen la etiqueta <?=SID?> adjunta a ellos; de otro modo, la sesión no estaría disponible en cada página PHP.

## Finalizar una sesión de usuario

Terminar una sesión es tan sencillo como crearla. Por ejemplo, el siguiente script PHP finalizará la sesión de usuario actual:

```

<?php
    session_start();
    $sessionArray=$HTTP_SESSION_VARS;
    session_destroy();

    foreach ($sessionArray as $session_name => $session_value) {
        unset($$session_name);
    }
    unset($sessionArray);
?>

```

Aquí es necesaria la directiva session\_start() para conectar la página script con la sesión actual. Entonces se asignan a \$sessionArray las variables de sesión almacenadas en la variable global \$HTTP\_SESSION\_VARS. Se llama a la función session\_destroy() para eliminar los datos de sesión

desde el servidor. El loop `foreach` elimina la asignación de cada variable de sesión. Finalmente la variable `$sessionArray` también pierde la asignación para eliminar totalmente la sesión.

**NOTA:** Por defecto, los datos de sesión están almacenados en archivos (porque `session.save_handler = files` en `php.ini`) y también por defecto los datos se almacenan en archivos de sesión (prefix `sess_identifier`) en el directorio `/tmp`. Si quiere cambiar el directorio de los archivos de sesión, determine que la directiva `session.save_path` en `php.ini` se dirija a un directorio distinto. Asegúrese de que el usuario del servidor es el único que puede leer los archivos de sesión. Sólo los puede leer el usuario del servidor.

## Utilizar MySQL con PHP

PHP funciona perfectamente con MySQL, Postgres, Oracle y con otras bases de datos. MySQL es la plataforma de bases de datos más utilizada por PHP. Si aún no ha instalado MySQL en su sistema, puede aprender a hacerlo en [www.mysql.com](http://www.mysql.com).

### Crear una página PHP sencilla para acceder a la base de datos MySQL

Para acceder a la base de datos MySQL necesita un nombre de usuario y una contraseña que acepte MySQL. Por ejemplo, el listado 15.11 muestra un script PHP llamado `simple_query.php`, que accede a una base de datos MySQL llamada `www` como usuario `httpd` con contraseña `nolsecret` desde el host local.

Listado 15.11. `simple_query.php`

```
<?php  
  
$host = 'localhost';  
$user = 'httpd';  
$passwd = 'nolsecret';  
$database_name = 'www';  
$table = 'users';  
  
$dbh = mysql_connect($host, $user, $passwd);  
  
mysql_select_db($database_name, $dbh);
```

```

?>
<html>
<head><title>Simple Query Script</title></head>
<body>

<table border=1>
<tr><th>Name</th> <th>Password</th></tr>

<?php

    $result = mysql_query("SELECT * from $table", $dbh);
    while ($myRow = mysql_fetch_row($result)) {
        printf("<tr><td>%s</td><td>%s</td></tr>", $myRow[0],
$myRow[1]);
    }
?>

</table>
</body>
</html>

```

A continuación tenemos lo que está ocurriendo en el listado anterior:

- Este script define las variables `$host`, `$user`, `$passwd`, `$database` y `$table` que se pueden asignar al nombre de host, al nombre de usuario, a la contraseña, al nombre de la base de datos y al nombre de la tabla, apropiados para el servidor MySQL.
- El script utiliza la función `mysql_connect()` para crear una conexión con el servidor de la base de datos cada vez que se solicita la página. El manejador de la conexión se almacena en otra variable llamada `$dbh`.
- La función `mysql_select_db()` selecciona la base de datos especificada por `$database` para utilizarla. En este momento, el manejador `$dbh` se conecta con el servidor de la base de datos y opera en dicha base de datos.
- El script imprime un documento HTML ordinario, que tiene otro pequeño script PHP embebido en él, el cual realiza una consulta SQL a la base de datos con la que ha conectado utilizando la función `mysql_query()`. La consulta se suministra como el argumento para esta función junto con el manejador de la base de datos (`$dbh`).
- Una vez que se realiza la consulta, el script utiliza el método `mysql_fetch_row()` para extraer filas de datos del conjunto de datos resultantes. Esta función devuelve un array de columnas para cada dato devuelto.
- Se utiliza una función `printf()` para imprimir dos elementos del array `$myRow`. Observe que el primer elemento del array `$myRow` está indexado con el 0.

7. El loop `while` continúa hasta que no tiene que pasar por más filas de `mysql_fetch_row`, al tiempo que el script se completa y el resto del documento HTML se imprime. La página resultante sería algo parecido a:

```
<html>
<head><title>Simple Query Script</title></head>
<body>

<table border=1>
<tr><th>Name</th> <th>Password</th></tr>

<tr><td>kabir</td><td>mysecret1</td></tr><tr><td>esmith</td><td>sale007</td></tr>
</table>
</body>
</html>
```

8. Como puede ver, se analiza el script PHP embebido, y el usuario del lado del cliente no sabe qué base de datos o qué usuario/ contraseña se ha utilizado para producir esta página.

Si tiene muchas páginas en las que utiliza la misma base de datos MySQL para mostrar elementos de datos utilizando varias consultas, debería utilizar la función `include()` para simplificar la gestión de tareas. Por ejemplo, el listado 15.12 muestra una versión modificada del script `simple_query.php` que incluye la función `include()`.

**Listado 15.12. simple\_query2.php**

```
<?php include('/usr/local/apache/secrets/mysql/header.inc'); ?>

<html>
<head><title>Simple Query Script</title></head>
<body>

<table border=1>
<tr><th>Name</th> <th>Password</th></tr>

<?php
    $sth = mysql_query("SELECT * from $table", $dbh);
    while ($myRow = mysql_fetch_row($sth)) {
        printf("<tr><td>%s</td><td>%s</td></tr>", $myRow[0],
$myRow[1]);
    }
?>

</table>
</body>
</html>
```

Cuando se accede a este script mediante la Web, la página resultante es la misma que la que produce simple\_query.php. Sin embargo, aquí el script utiliza un archivo de cabecera que contiene el script de conexión con la base de datos tal y como se muestra a continuación:

```
<?php  
  
$host = 'localhost';  
$user = 'httpd';  
$passwd = 'nolsecret';  
$database_name = 'www';  
$table = 'users';  
  
$dbh = mysql_connect($host, $user, $passwd);  
  
mysql_select_db($database_name, $dbh);  
  
?>
```

Eliminando este script de cada página que lo utiliza, facilita la posibilidad de cambiar el nombre del host, el nombre de usuario, la contraseña, el nombre de la base de datos y el nombre de la tabla. Si tiene 20 páginas utilizando la misma base de datos y la misma tabla, puede actualizar la contraseña en el archivo header.inc y lo tendrá resuelto.

## Asegurar archivos include PHP

Si utiliza archivos include, como se muestra en la última sección, para almacenar los datos que no quiere que vean usuarios, como son el nombre de usuario, la contraseña y otra información, como el nombre del host de la base de datos, el nombre de la base de datos y los nombres de las tablas, asegúrese de que mantiene los archivos include en un directorio seguro en el que los navegadores Web no puedan navegar. El mejor sitio para situarlos es fuera del árbol de documentos del sitio Web. Si la raíz de documentos es /www/mysite/htdocs, entonces tiene que crear un directorio llamado /www/mysite/secrets/mysql y mantener allí los archivos include.

Si tiene que crear los archivos include dentro de la raíz de documentos, desactive el navegador Web utilizando la siguiente configuración en httpd.conf:

```
<Directory /path/to/include_files>  
    <Limit>  
        order deny,allow  
        deny from all  
    </Limit>  
</Directory>
```

No olvide reemplazar /path/to/include\_files con la ruta del directorio real de los archivos include. Si mantiene sus archivos include en todo

el sitio Web, puede seguir desactivando el acceso Web a ellos utilizando el siguiente segmento de configuración en httpd.conf:

```
<Files ~ "\.inc$">
    Order allow,deny
    Deny from all
</Files>
```

**NOTA: Esto sólo funcionará si todos sus archivos include PHP están nombrados con la extensión .inc.**

## Autenticación de usuarios con PHP y MySQL

Puede utilizar un script PHP para autenticar usuarios mediante la base de datos MySQL. El listado 15.13 muestra un sencillo script que utiliza una base de datos MySQL para autenticar usuarios.

**Listado 15.13. auth.php**

```
<?php

ob_start();
include('/usr/local/apache/htdocs/mysql/header.inc');

function show_dialog($realm = "Restricted Section") {

    header("WWW-Authenticate: Basic realm='$realm'");
    header('HTTP/1.0 401 Unauthorized');
    echo 'Authorization Required.';
    exit;

}

if ((!isset($PHP_AUTH_USER)) || (!isset($PHP_AUTH_PW))) {

    show_challenge();

}

else if ((isset($PHP_AUTH_USER)) && (isset($PHP_AUTH_PW))) {

    $sth = mysql_query("SELECT 1 from $table WHERE
                           username = '$PHP_AUTH_USER'
                           and
                           passwd = '$PHP_AUTH_PW'",
                           $dbh);

    $success = mysql_fetch_row($sth);
```

```

if ($success[0] == '') {
    show_challenge();
} else {
    echo "<p>You're authorized!</p>";
    # haz algo más
}
}

ob_end_flush();
?>

```

Cuando se solicita este script, utiliza la función `isset()` para comprobar si están asignadas dos variables llamadas `$PHP_AUTH_USER` y `$PHP_AUTH_PW`. Estas dos variables están fijadas por PHP en el caso de que un usuario haya introducido un nombre de usuario y una contraseña en respuesta de un intento de autenticación básica.

Como la primera vez, el usuario no ha visto la caja de diálogo de autenticación, estas dos variables están vacías y se llama a la función `show_challenge()`. Esta función simplemente imprime las cabeceras de autenticación HTTP Basic, que fuerzan al navegador Web a desplegar una caja de diálogo que le pida al usuario que introduzca un nombre de usuario y una contraseña.

Cuando el usuario introduce un par con el nombre de usuario y la contraseña, el par se envía mediante la cabecera de respuesta de autenticación al servidor Apache, y PHP lo ve. PHP asigna entonces las variables `$PHP_AUTH_USER` y `$PHP_AUTH_PW` de forma acorde. Tras llamar al script de nuevo en la solicitud de autenticación, éste utiliza la base de datos MySQL para verificar si existe o no el par nombre de usuario/ contraseña. Si las credenciales del usuario (para nombre de usuario/ contraseña) son válidas, el script muestra el mensaje "You're authorized!" ("¡Esta autorizado!") y finaliza. Por otro lado, si las credenciales no son válidas, se reintenta la autenticación.

Observe que esa llamada `include('/usr/local/apache/htdocs/mysql/header.inc')` esconde todo el código de conexión con la base de datos. El archivo de cabecera `.inc` se muestra a continuación:

```

<?php

$host = 'localhost';
$user = 'httpd';
$passwd = 'nolosecret';
$database_name = 'www';
$table = 'users';

$dbh = mysql_connect($host, $user, $passwd);

```

```
    mysql_select_db($database_name, $dbh);  
?>
```

Aquí, la base de datos MySQL llamada `www` está en host local, y un usuario llamado `httpd` puede acceder a ella utilizando `nolsecret` como contraseña. La base de datos `www` tiene una tabla llamada `users`. Puede cambiar estos parámetros para conectar con la base de datos adecuada en el host local o en un servidor de bases de datos MySQL remoto. Este archivo incluye abre una conexión con dicha base de datos en el host apropiado y devolver un manejador de conexiones llamado `$dbh`, que está disponible para `auth.php`.

En el script `auth.php`, observe esta línea:

```
$sth = mysql_query("SELECT 1 from $table  
                      WHERE username = '$PHP_AUTH_USER' and  
                            passwd = '$PHP_AUTH_PW'",  
                      $dbh);
```

Esta línea realiza una consulta SQL que devuelve un 1 si las credenciales del usuario (almacenado en `$PHP_AUTH_USER` y en `$PHP_AUTH_PW` de forma automática por PHP) corresponden a los campos del nombre de usuario y contraseña, respectivamente, en la tabla (`users`).

**NOTA: Si la tabla users tiene un nombre distinto para el nombre de campo distinto username ó passwd, asegúrese de que cambia la sentencia para reflejar estos nombres.**

La sentencia `$success = mysql_fetch_row($sth)` devuelve un array llamado `$success`, que debería devolver el valor 1 como primer elemento en `$success[0]` si la consulta tiene éxito, o si el elemento está indefinido.

Utilizando el primer elemento `$success[0]`, se toma la decisión de mostrar (o no mostrar) el intento de autentificación utilizando `show_challenge()`.

Cuando se identifica un usuario, este sencillo script `auth.php` únicamente imprime un mensaje que indica que el usuario se ha autenticado con éxito. Por supuesto, puede hacer que este script haga muchas otras cosas. Por ejemplo, puede redirigir al usuario a un subdirectorio protegido utilizando `Header ("Location: /path/to/subdirectory");`





# 16 Utilizar Perl con Apache

---

## En este capítulo

1. Compilamos e instalamos `mod_perl`.
2. Ejecutamos scripts CGI mediante `mod_perl`.
3. Escribimos un módulo `mod_perl`.
4. Precargamos módulos.
5. Compartimos memoria.
6. Escribimos una aplicación de bases de datos en `mod_perl`.

El logro más importante del proyecto de integración de Apache y Perl fue traer todo el poder del lenguaje de programación Perl al servidor Apache. El resultado fue el desarrollo de `mod_perl`, que puede compilar y enlazar juntos Apache y Perl para proporcionar una interfaz Perl orientada a objetos al API del lenguaje C del servidor. Esto le permite a los programadores de Perl, escribir módulos Apache en Perl. Un módulo Apache-Perl podría funcionar durante las fases de manejador, analizador de cabeceras, traducción URL, autentificación, autorización, acceso, comprobación de escritura, ajustes, registro y limpieza de una solicitud.

El módulo mod\_perl se utiliza mucho con Apache para crear contenido dinámico. mod\_perl es un modo muy eficaz de utilizar Perl con Apache; no tendrá que volver a ejecutar scripts CGI basados en Perl, que son lentos, consumen recursos y no suelen resultar convenientes para sitios que reciben gran cantidad de solicitudes simultáneas por segundo. Este capítulo discute el modo de compilar, instalar y configurar Apache para mod\_perl, también describe cómo ejecutar sus scripts CGI antiguos utilizando mod\_perl, y cómo desarrollar módulos mod\_perl para sacar partido del API de Apache.

## Compilar e instalar mod\_perl

A continuación le indicamos el modo de bajar, compilar e instalar mod\_perl en su sistema:

1. Baje la última versión de la fuente de mod\_perl en el directorio /usr/local/src del sitio <http://perl.apache.org/dist> o del sitio mirror oficial.

**ADVERTENCIA:** Por favor, lea las notas de instalación y los archivos README de la distribución de la fuente antes de proceder a la compilación. Los pasos que se comentan a continuación podrían ser distintos en la última versión de mod\_perl y/o la última de Apache.

2. Como raíz, extraiga la distribución fuente de mod\_perl utilizando el comando tar xvzf mod\_perl-version.tar.gz. En esta sección se supone que ha extraído la distribución fuente de Apache en /usr/local/src.
3. Cambie al directorio /usr/local/src/mod\_perl-version y ejecute:

```
Perl Makefile.PL APACHE_SRC=../apache-version \
DO_HTTPD=1 \
USE_APACHE=1 \
PERL_MARK_WHERE=1 \
EVERYTHING=1
```

No olvide cambiar ../apache-version con el nombre de ruta apropiado.

4. Ejecute el comando make && make test && make install para compilar, probar e instalar la librería mod\_perl en la distribución fuente de Apache.

5. Cambie el directorio de la distribución fuente de Apache y ejecute `make install` para compilar e instalar Apache con soporte `mod_perl`.
6. Inicie el servidor Apache con la nueva compilación utilizando el comando `/usr/local/apache/bin/apachectl start`. Si está preparado para ejecutar una versión anterior del servidor Apache, utilice el comando `/usr/local/apache/bin/apachectl stop` para pararla y, entonces, ejecute el comando `start` para relanzar el nuevo servidor con las capacidades `mod_perl`. Si se encuentra en un sistema Unix, puede ejecutar el comando `lynx -dump -head http://localhost/` para descargar las cabeceras que muestra el servidor. Si `mod_perl` está bien instalado, verá información sobre la versión `mod_perl` en la información de cabecera.

## Ejecutar scripts CGI utilizando mod\_perl

Como los scripts CGI son lentos y consumen más recursos bajo alta carga, en un mundo ideal no ejecutaríamos scripts CGI cuando está funcionando `mod_perl` en el sistema. Sin embargo, la realidad es que los administradores de sistemas están muy ocupados y trasladar algo, que de hecho está funcionando, se suele obviar porque tienen otros asuntos más serios por los que preocuparse y una gran cantidad de trabajo. Afortunadamente, `mod_perl` le permite ejecutar sus scripts CGI utilizando un módulo `mod_perl` por defecto llamado `Apache::Registry.pm`. De ese modo, puede ejecutar sus scripts CGI bajo `mod_perl` inmediatamente. A continuación tenemos cómo hacerlo.

1. En el archivo `httpd.conf`, tiene que crear un alias llamado `/apps/` que señale a su directorio de scripts CGI añadiendo la línea siguiente:

```
Alias /apps/ "/www/mysite/cgi-bin"
```

Asegúrese de cambiar `/www/mysite/cgi-bin` al lugar donde esté situado el directorio de scripts CGI en su sistema.

2. Dígale a Apache que cargue el módulo `Apache::Registry` durante el arranque añadiendo la siguiente línea en `httpd.conf`:

```
PerlModule Apache::Registry
```

3. Dígale a Apache que ejecute todos los scripts mediante `Apache::Registry` para el directorio `/apps/` añadiendo el siguiente segmento de configuración en `httpd.conf`:

```
<Location /apps>
    SetHandler perl-script
    PerlHandler Apache::Registry
    Options ExecCGI
</Location>
```

4. Reinicie el servidor Apache utilizando el comando `/usr/local/apache/bin/apachectl restart`.
5. Acceda al script CGI utilizando un navegador Web con `http://your_server_name/apps/script_name`. Si tiene una directiva `ScriptAlias` asignada de modo que `/cgi-bin/` esté dirigido a `/www/mysite/cgi-bin` (o cualquiera que sea el nombre del directorio de scripts CGI en su sistema), entonces puede acceder a los scripts CGI como script "CGI" utilizando `http://your_server_name/cgi-bin/script_name`, o puede acceder al mismo script con `mod_perl` utilizando `http://your_server_name/apps/script_name`. La última opción tiene la ventaja de que no produce un nuevo proceso CGI para cada solicitud, permitiéndole una realización más rápida. Observe que la variable de entorno `mod_perl` puede distinguir cómo se está ejecutando un script (CGI o `mod_perl`). Considere, por ejemplo, el siguiente segmento de código:

```
if ($ENV{MOD_PERL} ne '') {  
  
    # Ejecute como script mod_perl, como un módulo mod_perl  
    # nativo o como el script Apache::Registry  
  
} else {  
  
    # El script CGI se está ejecutando mediante mod_cgi  
    # como un proceso separado  
}
```

La sentencia condicional anterior detecta cómo se está ejecutando un script. El script en el directorio `apps` se ejecutará mediante el módulo `Apache::Registry`. Esto significa que puede eliminar completamente el módulo `mod_cgi` de su sistema volviendo a compilar Apache con la opción `--disable-module=cgi`.

## No realice más trabajo del necesario

Antes de ponerse a escribir un buen módulo `mod_perl` para su servidor Web, busque en CPAN los módulos existentes, porque podrían resolver su problema. En el momento en el que se escribió este libro, había unos 500 módulos Perl específicos de `mod_perl` en CPAN. Puede ver los módulos disponibles en CPAN del siguiente modo:

1. Ejecute el comando `perl -MCPAN -e shell`.
2. Introduzca `i /Apache/` en el prompt de `cpan>`. Recibirá una lista de todos los módulos disponibles relacionados con Apache.

# Crear un módulo mod\_perl utilizando el API de Perl para Apache

Cuando instaló mod\_perl en su sistema, también instaló una poderosa interfaz del programa de aplicación (Application Programming Interface, API) Perl para Apache. Utilizando este API, puede desarrollar scripts mod\_perl que sacan partido de mod\_perl de un modo "nativo". Aunque puede ejecutar scripts CGI utilizando mod\_perl, estos no están diseñados para el mod\_perl, el cual utiliza el API de Perl y por lo tanto no puede sacar todo el partido del poder de mod\_perl. El siguiente material muestra cómo puede desarrollar scripts que se han escrito utilizando el API de Perl para Apache, para sacar todo el partido del módulo mod\_perl. Un script mod\_perl nativo está escrito como un módulo Perl con la arquitectura siguiente:

```
package MODULE_NAME;

sub handler {

    # Utiliza el objeto to do something useful de la solicitud
    # proporcionada por el API Apache
}

sub module_method_1 { # do something useful }
sub module_method_2 { # do something useful }
sub module_method_3 { # do something useful }
...
sub module_method_N { # do something useful }

1;
```

Aquí, el archivo del módulo empieza con un nombre de un paquete. El nombre del paquete debe corresponder al nombre del archivo del módulo. Por ejemplo, si nombras tu paquete (es decir, el módulo) como:

```
Package MyModule;
```

Entonces debe nombrar el archivo que contiene este módulo como `MyModule.pm`. Normalmente, los archivos de los módulos de Perl están guardados en un subdirectorio dentro de la lista de directorios que apunta al array `@INC`. Puede determinar cuáles son los directorios que apuntan a `@INC` ejecutando el siguiente comando desde la línea de comandos:

```
perl -le 'print join("\n", @INC)'
```

Debería ver una salida parecida a esta:

```
/usr/lib/perl5/5.6.0/i386-linux
/usr/lib/perl5/5.6.0
```

```
/usr/lib/perl5/site_perl/5.6.0/i386-linux  
/usr/lib/perl5/site_perl/5.6.0  
/usr/lib/perl5/site_perl
```

Si utiliza una versión de Perl distinta a 5.6.0, entonces la ruta debe reflejarlo. Cuando mod\_perl encuentra la primera solicitud para un módulo, busca en los directorios para localizar el módulo. Como no es una buena idea mezclar sus módulos personalizados con los módulos estándar proporcionados con Perl y con mod\_perl, tiene que crear un directorio nuevo llamado Development en /usr/lib/perl5/site\_perl y guardar sus módulos personalizados en este directorio. Esto permite a mod\_perl encontrar sus módulos personalizados en el directorio /usr/lib/perl5/site\_perl/Development cuando se necesitan. Además, asegúrese de asignar permisos de archivos para este directorio nuevo de modo que el usuario Apache (asignado por la directiva User en httpd.conf) pueda leer este directorio y los archivos que se encuentran en él. Puede utilizar los comandos siguientes para crear el directorio y asignar los permisos de archivo:

```
mkdir -p /usr/lib/perl5/site_perl/Development  
chown -R Apache_User:Apache_Group /usr/lib/perl5/site_perl/  
Development  
chmod -R 750 /usr/lib/perl5/site_perl/Development
```

No olvide cambiar Apache\_User y Apache\_Group con los nombres de usuario y de grupo adecuados tal y como se asignan en httpd.conf, utilizando las directivas User y Group, respectivamente.

Cuando tenga creado el directorio, puede crear un módulo sencillo como el que se muestra en el listado 16.1. Guarde este módulo como /usr/lib/perl5/site\_perl/Development/SimpleAPI.pm.

#### Listado 16.1. Un ejemplo de un API de Perl para Apache

```
package Development::SimpleAPI;  
  
use strict;  
  
use Apache::Constants qw(:common);  
  
my $callCounter = 0;  
  
sub handler {  
  
    my $r = shift;  
  
    $r->send_http_header('text/html');  
  
    print <<HTML_DOC;  
  
<html>
```

```

<head><title>Simple Perl API Example Script</title> </head>
<body>
    <h1>Simple Perl API Example Script</h1>
    <hr>
    <p>Process ID: $$ </p>
    <p>Request count: $callCounter </p>
    <hr>
</body>
</html>

HTML_DOC

$callCounter++;

return OK;

}

1;

```

El nombre del módulo es `Development::SimpleAPI`. Esto le dice a `mod_perl` que descargue el módulo desde el subdirectorio llamado `Development` bajo cualquiera de los directorios que apuntan a `@INC`. Como creó el subdirectorio `Development` en el directorio `/usr/lib/perl5/site_perl`, `mod_perl` cargará el módulo `SimpleAPI.pm` desde este directorio. El propósito de este módulo es demostrar las características `mod_perl`.

Este módulo funciona del siguiente modo:

- La línea `use strict;` le dice a Perl que fuerce la comprobación de las construcciones inseguras. Esta es una línea muy importante para los módulos `mod_perl`. Por ejemplo, si se utiliza una variable en el script sin declararla adecuadamente, utilizando una sentencia `my`, entonces Perl se quejará y le forzará a declararla antes de utilizarla.

**NOTA:** Este tipo de comprobación es muy importante porque los módulos `mod_perl` se cargan una vez en la memoria y se ejecutan hasta que el servidor se para. Si se utilizan una o más variables innecesarias, la posibilidad de que tenga lugar una filtración de memoria se incrementa.

- La utilización de la línea `Apache::Constants qw(:common);` simplemente carga el módulo `Constants.pm`, que define un conjunto de constantes, como `OK`, `REDIRECT` y `DONE`, que se pueden utilizar enseguida en el script.
- La siguiente línea define la variable `$callCounter` y la asigna el valor 0. Esta variable es global para el script y su valor está disponible para todas las invocaciones del script desde el proceso hijo de Apache. Además, observe que la variable está asignada a cero después de cargar el módulo.

- A continuación, se define el método `handler()`. Este método contiene el objeto de la solicitud Apache (`$r`) en forma de parámetro, el cual lo utiliza para escribir una cabecera Content-Type utilizando el método integrado `send_http_header()`.
- El método `handler()` imprime un documento HTML mínimo, que muestra el Process ID (`$$`) del proceso hijo de Apache y solicita un contador, utilizando la variable `$callCounter`. La variable `$callCounter` se incrementa después. El método vuelve con un estado `OK` y finaliza el proceso de la solicitud.
- La línea final `1;` es necesaria para los módulos Perl, que deben devolver un número distinto de cero para reunir los requisitos de un módulo.

Los siguientes pasos le muestran como puede ejecutar el módulo `SimpleAPI.pm`.

1. Primero, añada el siguiente segmento de configuración al archivo `httpd.conf`:

```
# Configuración para ejecutar los módulos mod_perl
PerlModule Development::SimpleAPI

<Location /firstone>
    SetHandler perl-script
    PerlHandler Development::SimpleAPI
</Location>
```

Lo que está ocurriendo en el segmento anterior es lo siguiente:

- La directiva `PerlModule` carga el módulo `Development::SimpleAPI` en el arranque del servidor Apache.
  - La directiva `<Location>` asigna un manejador del tipo `perl-script` para la localización `/firstone`.
  - La directiva `PerlHandler` le dice a Apache que el manejador `perl-script` (es decir, `mod_perl`) para la localización `/firstone` es el módulo `Development::SimpleAPI`, que podemos encontrar en el archivo `Development/SimpleAPI.pm` bajo un directorio señalado por el array `@INC`.
2. Reinicie el servidor Apache utilizando el comando `/usr/local/apache/bin/apachectl restart`.
  3. Ejecute el módulo `SimpleAPI.pm` por primera vez utilizando la URL `http://your_server_name/firstone`.

El contador de solicitudes empieza en el 0 y avanza a medida que actualiza la página. Además, el PID de los procesos hijo de Apache cambian con poca fre-

cuencia. Para asegurarse de que sabe cómo funcionan los módulos mod\_perl, debe entender que:

- Durante el arranque de cada proceso hijo obtiene acceso compartido a una copia del código SimpleAPI.pm. Esto elimina el coste CGI (es decir, el coste de invocar un script CGI cada vez que se solicita) porque el módulo ya está cargado en la memoria y se encuentra disponible para cada proceso hijo y para cada hilo (en caso de haberlos) de cada proceso hijo.
- La primera vez que accede a `http://your_server_name/firstone`, se ejecuta el módulo por primera vez y se incrementa la variable \$callCounter. Las siguientes llamadas (mediante la actualización de la pantalla del navegador) aumentan la cuenta de solicitudes. El contador sólo aumenta para el proceso hijo que está sirviendo su solicitud. Cuando otro proceso hijo inicia el contador, lo inicia en cero; \$callCounter no se comparte realmente entre los procesos hijo. En otras palabras, aunque el código del módulo se comparta entre los hijos Apache, no se comparten los datos.
- Cada vez que llamamos al módulo SimpleAPI.pm, Apache pasa una solicitud del objeto solicitado, llamado \$r, al método handler(). El método handler() actúa como punto de entrada para este módulo. El código que se encuentra fuera de este módulo, sólo se ejecuta si el método handler() le necesita directa o indirectamente. Por ejemplo, en el siguiente método, se llama a method\_one() y method\_two() porque handler() los necesita:

```
sub handler {  
  
    my $r = shift;  
  
    method_one();  
    method_two();  
  
    return OK;  
}  
  
sub method_one {  
    # somethig useful  
}  
  
sub method_two {  
    # somethig useful  
}
```

- El método handler() debe devolver un código de estado Apache como OK, DECLINED, DONE, NOT\_FOUND, FORBIDDEN, AUTH\_REQUIRED o SERVER\_ERROR. Puede encontrar una lista completa de constantes de

estados Apache en el archivo Constants.pm suministrado con mod\_perl. Utilice el comando locate Constants.pm para localizar el archivo, o utilice el buscador de archivos de su sistema para localizarlo. En los sistemas Linux que ejecutan Perl 5.6.0, la ruta del archivo es /usr/lib/perl5/site\_perl/5.6.0/i386-linux/Apache/Constants.pm.

## Utilizar CGI.pm para escribir módulos mod\_perl

La mayoría de la gente que utiliza Perl con Apache sabe cómo utilizar el módulo CGI.pm para escribir scripts CGI en Perl. Por suerte, el autor de CGI.pm se dio cuenta de que la gente que suele utilizar este módulo podría seguir utilizándolo en un entorno que no fuese de CGI, como mod\_perl, de modo que el autor del módulo diseñó este módulo para que fuese muy fácil de utilizar bajo mod\_perl. Por ejemplo, el listado 16.2 muestra una versión de CGI.pm para el módulo SimpleAPI.pm, que ahora se llama SimpleAPIUsingCGI.pm.

**Listado 16.2.** Un ejemplo sencillo que utiliza un módulo CGI.pm en un módulo mod\_perl

```
package Development::SimpleAPIUsingCGI;

use strict;

use Apache::Constants qw(:common);

use CGI (-compile: all);

my $callCounter = 0;

sub handler {

    my $query = new CGI;
    print $query->header;
    print $query->start_html('Simple Perl API Example Using CGI.pm'),
        $query->h1('Simple Perl API Example Using CGI.pm Module'),
        $query->hr,
        $query->p("Process ID: $$"),
        $query->p("Request count: $callCounter"),
        $query->hr,
        $query->end_html;

    $callCounter++;

    return OK;
}

1;
```

Los desarrolladores de CGI.pm se dieron cuenta de que yo utilizaba la opción `-compile`: all cuando le decía a Perl que quería utilizar el módulo CGI.pm. Le estamos pidiendo a Perl que compile todo el código CGI.pm en la carga inicial del módulo durante el arranque del servidor. Esto garantiza que todas las características de CGI.pm están realmente disponibles para el módulo SimpleAPIUsingCGI.pm. No será necesario compilar más veces ningún código específico de CGI.pm durante el ciclo de una solicitud. Por supuesto, si no tiene que utilizar muchas características de CGI.pm, puede utilizar CGI qw(`-compile :standard`), o CGI qw(`-compile :standard :html3`), y similares, para reducir la utilización de memoria por parte del CGI.pm. Recuerde que mantener una gran cantidad de código inutilizado en memoria gasta los recursos del sistema.

Además, observe que el objeto CGI, \$query, se crea dentro del método `handler()`. Esto es necesario porque si crea un objeto CGI fuera del método `handler()` en la sección global del módulo, como en el caso de \$callCounter, se creará el objeto una vez para la primera solicitud y sólo tendrá esa información de la solicitud. Por ejemplo, si mueve la línea `my $query = new CGI;` fuera del método `handler()` y la coloca justo detrás de la línea `my $callCounter = 0;` el script va a tratar cada una de las siguientes solicitudes como si fuera la misma solicitud. Por eso, si necesita acceder a la información de la cadena de consulta, utilizando el método `$query->param()`, sólo obtendrá el dato de la primera solicitud, porque no se ha creado el objeto \$query para cada solicitud. Este es el motivo por el cual es muy importante que cree el objeto \$query específico de solicitud en el método `handler()` tal y como se muestra en el listado 16.2. El módulo SimpleAPIUsingCGI.pm proporciona la misma funcionalidad que SimpleAPI.pm y se puede ejecutar mediante la URL `http://your_server_name/cgipm_example` tras añadir las siguientes líneas a `httpd.conf` y reiniciar el servidor:

```
# Configuración para ejecutar los módulos mod_perl
PerlModule Development::SimpleAPIUsingCGI

<Location /cgipm_example>
    SetHandler perl-script
    PerlHandler Development::SimpleAPIUsingCGI
</Location>
```

## Precargar módulos Perl para ahorrar memoria

Si utiliza muchos módulos CPAN como CGI o DBI, o tiene muchos módulos personalizados para su servidor Apache, podría utilizar la directiva `PerlModule` para precargarlos durante el arranque del servidor. Haciendo esto ahorrará tiempo durante la primera solicitud de esos módulos y además aumentará la posibilidad de compartir las páginas de código (memoria) entre los procesos hijo de

Apache. Por ejemplo, si utiliza los módulos estándar CGI, DBI y Digest::MD5 en uno o más de sus módulos mod\_perl, puede precargar estos módulos colocando las siguientes directivas en httpd.conf:

```
PerlModule CGI  
PerlModule DBI  
PerlModule Digest::MD5
```

Otra aproximación es utilizar la directiva PerlRequire. Esta directiva se puede asignar a un script Perl externo, el cual cargue todos los módulos que quiera precargar y posiblemente compartir entre los procesos hijo de Apache. Por ejemplo:

```
PerlRequire /usr/local/apache/apps/startup.pl
```

Indica que Apache necesita el script /usr/local/apache/apps/startup.pl durante el arranque. Este script se carga en la memoria, que entonces, debería cargar los módulos necesarios. Por ejemplo, este script podría ser:

```
#!/usr/bin/perl  
  
use CGI ();  
use DBI ();  
use Digest::MD5 ();  
1;
```

Cada una de las líneas module\_name () utilizadas, cargan un módulo determinado. Los paréntesis vacíos garantizan que esa importación por defecto de símbolos desde estos módulos no se está realizando, lo que ahorra algo de memoria.

**NOTA:** Si utiliza todas las características del módulo CGI.pm puede añadir CGI->compile(':all') al script startup.pl para compilarlo, lo que ahorrará tiempo durante el proceso de solicitudes.

Además, asegúrese de que el usuario Apache (asignado utilizando la directiva User en el archivo httpd.conf) puede acceder y ejecutar el script startup.pl.

## Seguir la pista de los módulos mod\_perl en la memoria

Ser capaz de seguir la pista de qué módulo mod\_perl está utilizando el servidor Web, es una gran ayuda para la administración del sistema. Utilizando el módulo Apache::Status contenido en mod\_perl, puede seguir la pista de

los módulos cargados y obtener mucha información sobre ellos. Para hacerlo, siga los pasos siguientes:

1. Como raíz, debe instalar el módulo `Devel::Syndump` de un sitio mirror CPAN utilizando este comando:

```
perl -MCPAN -e 'install Devel::Syndump'
```

2. Añada el siguiente segmento de configuración al `httpd.conf` antes de cualquier directiva `PerlModule`.

```
PerlModule Apache::Status

<Location /perl-status>
    SetHandler perl-script
    PerlHandler Apache::Status
</Location>
```

Este código asegura que el módulo `Apache::Status` se ha cargado antes de cualquier otro módulo. Si utiliza la directiva `PerlRequire` para cargar módulos mediante un script de arranque externo, no necesita la línea `PerlModule Apache::Status`. En lugar de esto, cargue el módulo `Apache::Status` en el script de arranque antes de cualquier otro módulo.

3. Reinicie el servidor Apache y acceda a la información de los módulos `mod_perl` cargados utilizando el estado `http://your_server_name/perl`. Esta página muestra una lista de opciones (enlaces), como la que se muestra a continuación:

```
Perl Configuration
Loaded Modules
Inheritance Tree
Enabled mod_perl Hooks
Environment
PerlRequired Files
Signal Handlers
Symbol Table Dump
ISA Tree
Compiled Registry Scripts
```

4. Haga clic en el enlace `Loaded Modules` para ver todos los módulos `mod_perl` cargados. Los otros enlaces también proporcionan importante información de estado sobre las especificaciones de `mod_perl`.

## Implementar ASP utilizando el módulo Apache::ASP

Active Server Page (ASP) es una plataforma común de script en el mundo Windows. Sin embargo, va a dejar de ser una plataforma sólo para Windows.

Utilizando mod\_perl con el módulo Apache::ASP CPAN puede crear páginas ASP que embeban scripts ASP basados en Perl. A continuación tiene cómo:

1. Como raíz, ejecute el comando siguiente:

```
perl -MCPAN -e 'install Apache::ASP'
```

El módulo CPAN instala el módulo Apache::ASP desde un sitio mirror CPAN. Si falla por alguna razón, investigue por qué y resuelva los problemas. Normalmente la instalación tiene éxito a no ser que se olvide de algún módulo necesario. En ese caso, instale el módulo del mismo modo e instale Apache::ASP.

2. Una vez que tiene instalado el módulo Apache::ASP, tiene que crear un subdirectorio llamado `asp` en su directorio raíz de documentos. Supongo que su directiva `DocumentRoot` está asignada a `/www/mysite/htdocs` y que ha creado el directorio `/www/mysite/htdocs/asp`.
3. Tiene que crear un script de prueba llamado `test.asp`, tal y como se muestra a continuación, y almacenarlo en el directorio `/www/mysite/htdocs/asp`.

```
<html>
<head><title>ASP Example</title></head>
<body>

<%
    $Application->Lock();
    $Application->{Count}+=1;
    $Application->UnLock();

%>

<h1> ASP Page </h1>
<hr>
<p>
This page has been called <=%=$Application->{Count}%> times.

</body>
</html>
```

esta página ASP `test.asp` simplemente incrementa en 1 el objeto de la aplicación ASP, cada vez que se llama a esta página.

4. Añada las líneas siguientes al archivo `httpd.conf`:

```
Alias /asp/ "/www/mysite/htdocs/asp"

<Location /asp/>
    SetHandler perl-script
    PerlHandler Apache::ASP
    PerlSetVar Global /tmp
</Location>
```

Esto es lo que está ocurriendo en las líneas anteriores:

- El alias /asp/ señala al directorio /www/mysite/htdocs/asp. Cámbielo para que apunte al directorio apropiado de su sistema.
  - El contenedor <Location> le dice a Apache que utilice el módulo Apache::ASP para manejar todos los archivos de este directorio; también asigna la variable Global al directorio /tmp.
  - El módulo Apache::ASP crea un directorio llamado /tmp/.state en el que almacena la información de estado de modo que incluso si apaga y reinicia el servidor, no se pierde la información de estado.
5. Cambie los permisos de archivo para el directorio asp y su contenido, para permitirle al usuario Apache (es decir, cualquiera que haya asignado para la directiva User en httpd.conf) que acceda a los archivos. Por ejemplo, utilice los comandos chown -R httpd:httpd /www/mysite/htdocs/asp && chmod -R 750 /www/mysite/htdocs/asp para asegurar que ese usuario Apache (httpd) y el grupo Apache (httpd), pueda acceder al directorio y a los archivos almacenados en él.
6. Reinicie el servidor Apache utilizando el comando /usr/local/apache/bin/apachectl restart y acceda a la página test.asp utilizando http://your\_server\_name/asp/test.asp.

**NOTA:** Para aprender más cosas sobre cómo escribir scripts ASP en Perl, visite el sitio Web ASP en [www.apache-asp.org](http://www.apache-asp.org).



# 17 Ejecutar servlets de Java y páginas JSP con Tomcat

---

## En este capítulo

1. Instalamos JDK.
2. Instalamos Tomcat.
3. Instalamos el módulo `mod_jk` para Apache.
4. Configuramos Tomcat para Apache.
5. Configuramos servlets para Apache.
6. Configuramos nuestros propios servlets o páginas JSP.
7. Configuramos Tomcat para utilizar Java Security Manager.
8. Ejecutamos servlets Java mediante Tomcat.
9. Desactivamos el servicio HTTP Tomcat.

En los últimos años, Java se ha convertido en el líder de las plataformas de aplicaciones Web para las necesidades de desarrollo en el ámbito de empresa. Muchas grandes compañías rechazan la idea de utilizar otra plataforma que no sea Java. Afortunadamente, Apache puede funcionar muy bien con Java. Utilizan-

do un adaptador (módulo) de Apache, éste puede interaccionar con Tomcat, la implementación oficial de referencia para Java servlet 2.2 y para JSP 1.1.

Un servlet es un programa Java que se ejecuta dentro de la máquina virtual de Java (JVM) en el sistema del servidor Web. Cuando se recibe una solicitud para un servlet, se crea un hilo nuevo desde el servlet que se ha cargado. Por eso, un servlet tiene muchos menos requisitos de arranque que los que tiene un script CGI. Al igual que un programa CGI, un servlet puede interaccionar con solicitudes HTTP desde navegadores Web, y con bases de datos relacionales como Oracle, Postgres y MySQL, o con otras aplicaciones externas, y entonces escribir resultados en una respuesta HTTP. Un servlet tiene acceso a datos suministrados por el cliente, cabeceras HTTP como cookies, información sobre el navegador, información sobre el host, y similares, al igual que un programa CGI.

Tomcat, un entorno Java de código abierto (llamado contenedor), para servlets de Java, maneja solicitudes de servlets, pero hay un pacto que se puede establecer entre Tomcat y Apache. Puede utilizar un módulo de Apache llamado `mod_jk` (antes se llamaba `mod_jserv`) para interaccionar con Tomcat de manera que Tomcat puede ser responsable de las solicitudes servlet y Apache puede manejar el resto. Tomcat también implementa soporte Java Server Page (JSP) que le permite colocar código Java en páginas HTML con extensiones `.jsp`. El servlet analiza las páginas con estas extensiones de forma parecida a como se analizan las páginas PHP o ASP. Esto hace las JSP muy parecidas a PHP o a Active Server Pages (ASP), con la ventaja añadida de tener la transportabilidad de Java y la robustez de JSP Application Programming Interface (API). Realmente, JSP tiene todas las capacidades de los servlets de Java, pero para pequeñas tareas de programación, es más fácil de escribir que el código Java. Tomcat convierte automáticamente los scripts JSP en servlets cuando se han referenciado, compilados e iniciado primero en el contenedor del servlet.

Este capítulo discute cómo hacer que Tomcat y `mod_jk` funcionen con Apache de modo que pueda utilizar servlets y JSP.

**NOTA:** Si no está familiarizado con los servlets de Java o con JSP, puede leer un libro sobre esto antes de adentrarse en este capítulo. Una discusión detallada sobre servlets o JSP está fuera del alcance de este libro.

## Utilizar servlets

Las ventajas de ejecutar servlets con respecto a los scripts CGI, o incluso aplicaciones `mod_perl` o FastCGI, son:

- Los servlets son realmente independientes de plataforma y por lo tanto muy transportables. Todas las plataformas modernas de servidores sopor-

tan Java. Por lo tanto, desplegar servlets en una red híbrida es mucho más fácil que cualquier otra plataforma de aplicaciones Web.

- Experimenta menor sobrecarga que un programa CGI debido a que tiene una huella de arranque menor. No se necesitan producir procesos nuevos para servir una solicitud. Simplemente se crea un hilo nuevo para servir la nueva solicitud, lo que tiene mucho menor coste de recursos que iniciar un nuevo proceso.
- La conectividad a bases de datos mediante Java Database Connectivity (JDBC) es preferible a la conectividad mediante el DBI Perl, que tiene problemas de rendimiento importantes en entornos de alta carga. Los scripts CGI conectan con la base de datos cada vez que arrancan; los scripts mod\_perl pueden meter en el caché la conexión con la base de datos pero no es un buen modo de gestionar conexiones. Un servidor Apache ejecutando 50 hijos puede potencialmente abrir 50 conexiones con la base de datos. Como los sitios de alto rendimiento tienen muchos servidores Web, la mayoría de las RDBM disponibles hoy en día, serán incapaces de gestionar este tipo de requisitos para la conexión de bases de datos. La mejor aproximación es utilizar un grupo de conexiones, que en un entorno mod\_perl tienen una gran cantidad de código personalizado pero es muy simple en un entorno servlet.
- Los servlets de Java se ejecutan en el "buzón" Java (JVM), el cual los convierte en una opción más segura que otras plataformas de aplicaciones Web, ya no hay que preocuparse más sobre la ejecución del buffer.

## Instalar Tomcat

Para instalar Tomcat en su servidor, tiene que tener un Java Run-time Environment (JRE), y si ha pensado desarrollar servlets, entonces necesitará también el Java Development Kit (JDK). Además, para utilizar Tomcat con Apache necesita el módulo mod\_jk. En esta sección veremos cómo instalar estos componentes para crear un entorno Tomcat.

## Instalar el último JDK para Tomcat

Antes de instalar Tomcat, necesita un Java Runtime Environment (JRE) apropiado. Necesita un kit completo de desarrollo de software (SDK) si va a escribir y compilar sus propios servlets de Java. El Java Development Kit (JDK), disponible gratuitamente en Sun Microsystems (<http://java.sun.com/>), incluye tanto JRE como paquetes de desarrollo. Puede instalar JDK para un sistema Linux siguiendo los siguientes pasos:

1. Baje la última versión de JDK (es decir, SDK) del sitio Web oficial de Java. En el momento en el que se escribió este libro, la última versión era 1.3 y se ofrecía como un script de shell de auto extracción que creaba un paquete RPM binario. Puede bajar el j2sdk-1\_3\_0\_02-linux-rpm.bin (o la última versión disponible) en un directorio.

2. Como raíz, ejecute:

```
sh j2sdk-version-linux-rpm.bin
```

por ejemplo, para la versión 1.3 puede ejecutar:

```
sh j2sdk-1_3_0_02-linux-rpm.bin
```

Cuando este script se ejecuta muestra una licencia, que tiene que confirmar. El script desempaquetá y crea un paquete RPM llamado j2sdk-version-linux.rpm (para JDK 1.3 es j2sdk-1\_3\_0\_02-linux.rpm).

3. Ejecute rpm -ivh j2sdk-version-linux.rpm para instalar el JDK. Por ejemplo, para instalar el JDK 1.3, ejecute el comando rpm -ivh j2sdk-1\_3\_0\_02-linux.rpm.
4. El JDK se instala en el directorio /usr/java/jdkversion. Por ejemplo, el 1.3 JDK está instalado en el directorio /usr/java/jdk1.3.0\_02. Este es el directorio JAVA\_HOME. Debería establecer una variable de entorno llamada \$JAVA\_HOME para señalar a este directorio. Si utiliza el shell bash, puede asignarlo en su archivo ~/.bashrc del siguiente modo:

```
export JAVA_HOME=/usr/java/jdkversion
```

Por ejemplo:

```
export JAVA_HOME=/usr/java/jdk1.3.0_02
```

Introduzca echo \$JAVA\_HOME en su prompt de shell para determinar si esta variable de entorno está asignada. Si la ha añadido al archivo .bashrc, entonces debe ejecutar source ~/.bashrc para exportar esta variable.

**NOTA: Si utiliza un shell distinto, como tcsh o csh, utilice el comando setenv JAVA\_HOME /usr/java/jdkversion en lugar del comando export.**

5. Añada el directorio \$JAVA\_HOME en su variable de entorno \$PATH utilizando la línea siguiente en su archivo ~/.bashrc.

```
export PATH=${PATH}: ${JAVA_HOME}/bin
```

Esta línea debe seguir la línea JAVA\_HOME discutida en el último paso.

6. Pruebe la instalación JDK ejecutando el comando `java -version`. Debería ver una salida parecida a la siguiente:

```
java version "1.3.0_02"
Java(TM) 2 Runtime Environment, Standard Edition (build
1.3.0_02)
Java HotSpot(TM) Client VM (build 1.3.0_02, mixed mode)
```

## Instalar Tomcat y el módulo mod\_jk

Una vez que tiene instalada la última versión de JDK desde sitio Web oficial de Java (tal y como se describió en la última sección), está preparado para instalar Tomcat y el módulo mod\_jk del siguiente modo:

Baje los últimos paquetes binarios de Tomcat del sitio Web oficial de Tomcat en <http://jakarta.apache.org>. En el momento en el que se escribía este libro, la última versión de Tomcat era 3.2, que habrá cambiado a 4.0 en el momento de su puesta en el mercado. Asegúrese de cambiar los números de las versiones mencionadas en las instrucciones que se dan a continuación.

Para Linux, los últimos paquetes RPM binarios se pueden bajar del directorio <http://jakarta.apache.org/builds/jakarta-tomcat/release/v3.2.1/rpms>. Los paquetes que debería bajar para Linux son:

- `tomcat-version.noarch.rpm` (versión actual: `tomcat-3.2.1-1.noarch.rpm`). Esta es la distribución binaria de Tomcat.
- `tomcat-mod-version.i386.rpm` (versión actual: `tomcat-mod-3.2.1-1.i386.rpm`). Contiene la distribución binaria de mod\_jk y los módulos mod\_jserv más antiguos para Apache. Instalará el módulo `mod_jk.so` en el directorio `/usr/lib/apache`. Como es un módulo Dynamic Shared Object (DSO) de Apache, debe tenerlo Apache compilado con soporte compartido de objetos. Puede comprobar rápidamente si el binario de Apache que está ejecutando, tiene soporte `mod_so`. Ejecute el comando `/usr/local/apache/bin/httpd -l` para realizar una lista con todos los módulos que están compilados en el binario de Apache. Si `mod_so.c` no se encuentra en la lista, tiene que volver a compilar Apache utilizando la opción `--enable-module=so` en el script `configure` que se encuentra en la distribución de la fuente Apache, y entonces ejecutar el comando `make && make install` para reinstalar Apache con soporte DSO.
- `tomcat-manual-version.noarch.rpm` (versión actual: `tomcat-manual-3.2.1-1.noarch.rpm`). Este es el paquete de documentos de Tomcat.

Baje todos los archivos anteriores en un directorio temporal nuevo y ejecute el comando `rpm -ivh tomcat*.rpm` desde ese directorio como raíz para instalar todo el software.

Por defecto, Tomcat se instala en el directorio /var/tomcat. Tiene que añadir la siguiente variable de entorno en su archivo ~/.bashrc:

```
export TOMCAT_HOME=/var/tomcat
```

Cargue la variable en su shell actual ejecutando el comando source ~/ .bashrc.

## Configurar Tomcat

En esta sección veremos cómo se configura Tomcat para Apache y cómo gestionar seguridad para Tomcat utilizando la herramienta Java Security Manager.

### Configurar Tomcat para Apache

El archivo de configuración principal para Tomcat es /var/tomcat/server.xml. Como se puede imaginar por su extensión, se trata de un archivo XML.

Tomcat puede utilizar dos tipos de manejadores de conexión, los protocolos Ajp13 y Ajp12. El protocolo Ajp13 es el protocolo más reciente, y proporciona un rendimiento más alto, y además soporta conectividad Secure Socket Layer (SSL).

Para activar el manejador de conexión basado en el protocolo Ajp13 en Tomcat, tiene que asegurarse de que la siguiente configuración del conectador no está comentada en el archivo server.xml:

```
<Connector  
    className="org.apache.tomcat.service.PoolTcpConnector">  
  
<Parameter name="handler"  
    value="org.apache.tomcat.service.connector.  
Ajp13ConnectionHandler"/>  
  
    <Parameter name="port" value="8009"/>  
</Connector>
```

**ADVERTENCIA:** El archivo servlet.xml debería tener ya un bloque de código parecido al anterior para las conexiones Ajp12 en el puerto 8007. Incluso si piensa utilizar el conectador basado en Ajp13, no debería eliminar este conectador. Se necesita para apagar Tomcat.

Las propiedades de una instancia Tomcat, llamada worker, están definidas en el archivo /var/tomcat/conf/workers.properties. El archivo por defecto (sin comentarios) contiene las propiedades que se muestran en el listado 17.1.

### Listado 17.1. Propiedades de Tomcat por defecto (worker)

```
workers.tomcat_home=c:\jakarta-tomcat
workers.java_home=c:\jdk1.2.2
ps\
worker.list=ajp12, ajp13

worker.ajp12.port=8007
worker.ajp12.host=localhost
worker.ajp12.type=ajp12
worker.ajp12.lbfactor=1

worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
worker.ajp13.lbfactor=1

worker.loadbalancer.type=lb
worker.loadbalancer.balanced_workers=ajp12, ajp13

worker.inprocess.type=jni
worker.inprocess.class_path=$(workers.tomcat_home)$(ps)classes

worker.inprocess.class_path=$(workers.tomcat_home)$(ps)
lib$(ps)jaxp.jar
worker.inprocess.class_path=$(workers.tomcat_home)$(ps)
lib$(ps)parser.jar

worker.inprocess.class_path=$(workers.tomcat_home)$(ps)lib$
(ps)jasper.jar
worker.inprocess.class_path=$(workers.tomcat_home)$(ps)lib$
(ps)servlet.jar
worker.inprocess.class_path=$(workers.tomcat_home)$(ps)lib$
(ps)webserver.jar

worker.inprocess.class_path=$(workers.java_home)$(ps)lib$
(ps)tools.jar

worker.inprocess.cmd_line=-config
worker.inprocess.cmd_line=$(workers.tomcat_home)/conf/
jni_server.xml
worker.inprocess.cmd_line=-home
worker.inprocess.cmd_line=$(workers.tomcat_home)

worker.inprocess.jvm_lib=$(workers.java_home)$(ps)jre$(ps)bin$(
ps)classic$(ps)jvm.dll

worker.inprocess.stdout=$(workers.tomcat_home)$(ps)
inprocess.stdout
worker.inprocess.stderr=$(workers.tomcat_home)$(ps)
inprocess.stderr
worker.inprocess.sysprops=tomcat.home=$(workers.tomcat_home)
```

Este archivo de propiedades por defecto define un worker ajp12, un worker ajp13, un worker jni y un worker lb. Las diferencias entre estos tipos de worker se muestran en la tabla 17.1.

**Tabla 17.1.** Diferencias entre los tipos de Workers

Tipo de worker	Descripción
ajp12	Este worker utiliza el protocolo ajpv12 para dirigir solicitudes a workers Tomcat fuera de proceso que están utilizando el protocolo ajpv12.
ajp13	Este worker utiliza el protocolo ajpv13 para dirigir solicitudes a workers Tomcat fuera de proceso que están utilizando el protocolo ajpv12. Como ajpv13 es un protocolo más nuevo que ajpv12, el tipo ajp13 de worker puede alcanzar mayor rendimiento que el worker ajp12. Además, ajp13 ofrece soporte SSL.
jni	Utilizando Java Network Interface (JNI), este tipo de worker puede dirigir solicitudes a workers Tomcat fuera de proceso. Un worker en proceso se ejecuta dentro de un JVM en el propio espacio de memoria de Apache.
lb	Utilizando un sencillo esquema de equilibrio de carga con retorno al punto de origen, este worker puede dirigir solicitudes.

El archivo `workers.properties` tiene unas cuantas características que son "globales" para todos los tipos de workers. Por ejemplo:

```
workers.tomcat_home=c:\jakarta-tomcat  
workers.java_home=c:\jdk1.2.2  
ps=\  
worker.list=ajp12, ajp13
```

La primera determina el directorio local para Tomcat. Debe estar asignado al valor de la variable de entorno `$TOMCAT_HOME`. Como la instalación por defecto de Tomcat crea `/var/tomcat` como el directorio bajo el sistema Linux, debería cambiar el valor por defecto (`c:\Jakarta-tomcat`) del siguiente modo:

```
workers.tomcat_home=/var/tomcat
```

Del mismo modo, el `workers.java_home` asigna la ruta de máximo nivel para el directorio de instalación de JDK, que bajo Linux es el `/usr/java/jdkversion` (por ejemplo, para JDK 1.3, es `/usr/java/jdk1.3.0_02`). De este modo, debería asignarse:

```
workers.java_home=/usr/java/jdk1.3.0_02
```

La siguiente línea le dice a Tomcat qué separador utilizar para separar elementos de directorio. El valor por defecto \ funciona en los sistemas Windows, porque los nombres de directorios en las plataformas Windows se separan utilizando \. En los sistemas UNIX, debería ser /. Por eso, asigne esta línea como se muestra a continuación:

```
ps=/
```

El worker.list le dice a Tomcat el número de workers que quiere definir. La lista por defecto define ajp12 y ajp13 como dos workers.

Cada uno de estos workers puede tener sus propias propiedades asignadas utilizando las líneas worker.<worker\_name>.properties = value. Por ejemplo, en el archivo por defecto, el worker ajp13 tiene asignado los atributos siguientes:

```
worker.ajp12.port=8007  
worker.ajp12.host=localhost  
worker.ajp12.type=ajp12  
worker.ajp12.lbfactor=1
```

La primera línea asigna el número de puerto, que utiliza el worker ajp12 para escuchar solicitudes; la segunda define el nombre del host en el que este worker está escuchando conexiones y la tercera define el tipo de worker. Considero una decisión equivocada nombrar al worker como ajp12, ya que también es del tipo ajp12. Puede nombrar un worker como quiera siempre que consista en letras y números.

La cuarta línea le dice al worker cuál es la carga media para propósitos de equilibrio de carga. Un número alto indica una máquina potente y cuando hay muchas máquinas implicadas en un escenario de equilibrio de carga, será el worker del equilibrio de carga el que elija el worker, el cual tendrá el mayor lbfactor.

En las siguientes cuatro líneas se definen las mismas propiedades para el worker ajp13 en el archivo workers.properties.

Las siguientes dos líneas definen un worker llamado loadbalancer, que es del tipo lb y el cual equilibra los workers ajp12 y ajp13 utilizando un esquema de retorno al punto de origen. Estas dos líneas son:

```
worker.loadbalancer.type=lb  
worker.loadbalancer.balanced_workers=ajp12, ajp13
```

Las siguientes dos líneas definen un worker llamado inprocess del tipo jni. La class\_path del worker que está definida asignando la segunda línea a /var/tomcat/classes para un sistema Linux:

```
worker.inprocess.type=jni  
worker.inprocess.class_path=$(workers.tomcat_home)$(ps)classes
```

Las siguientes seis líneas añaden seis archivos de librerías a la ruta class:

```
worker.inprocess.class_path=$(workers.tomcat_home)$(ps)lib$(ps)jaxp.jar
```

```
worker.inprocess.class_path=$(workers.tomcat_home)$ (ps) lib$  
 (ps)parser.jar  
worker.inprocess.class_path=$(workers.tomcat_home)$ (ps) lib$  
 (ps)jasper.jar  
worker.inprocess.class_path=$(workers.tomcat_home)$ (ps) lib$  
 (ps)servlet.jar  
worker.inprocess.class_path=$(workers.tomcat_home)$ (ps) lib$  
 (ps)webserver.jar  
worker.inprocess.class_path=$(workers.java_home)$ (ps) lib$  
 (ps)tools.jar
```

La ruta class final (en Linux) para el worker inprocess es:

```
/var/tomcat/classes:/var/tomcat/lib/jaxp.jar:/var/tomcat/lib/  
parser.jar:/var/tomcat/lib/jasper.jar:/var/tomcat/lib/  
servlet.jar:/var/tomcat/lib/webserver.jar:/usr/java/  
jdk1.3.0_02/lib/tools.jar
```

Las siguientes cuatro líneas definen un conjunto de opciones de línea de comando para el worker inprocess:

```
worker.inprocess.cmd_line==config  
worker.inprocess.cmd_line=$(workers.tomcat_home)/conf/  
jni_server.xml  
worker.inprocess.cmd_line==home  
worker.inprocess.cmd_line=$(workers.tomcat_home)
```

La siguiente línea define la ruta de la librería JVM:

```
worker.inprocess.jvm_lib=$(workers.java_home)$ (ps) jre$(ps)bin$  
(ps)classic$(ps)jvm.dll
```

La ruta de la librería JVM (en Linux) es /usr/java/jdk1.3.0\_02/jre/lib/i386/classic/libjvm.so. De modo, que tiene que cambiar la línea jvm\_lib anterior:

```
worker.inprocess.jvm_lib=$(workers.java_home)$ (ps) jre$(ps)lib$  
(ps)i386$(ps)classic$(ps)libjvm.so
```

Las siguientes dos líneas definen los nombres de archivo que se están utilizando para escribir el STDOUT y el STDERR para el worker inprocess:

```
worker.inprocess.stdout=$(workers.tomcat_home)$ (ps)  
inprocess.stdout  
worker.inprocess.stderr=$(workers.tomcat_home)$ (ps)  
inprocess.stderr
```

El STDOUT está escrito en /var/tomcat/inprocess.stdout y el STDERR está escrito en /var/tomcat/inprocess.stderr en un sistema Linux.

La línea final define una propiedad del sistema para el worker inprocess. La propiedad por defecto asignada por la siguiente línea es tomcat.home con el valor /var/tomcat en Linux:

```
worker.inprocess.sysprops=tomcat.home=${workers.tomcat_home}
```

## Configurar Tomcat para utilizar el Java Security Manager

El Java Security Manager hace cumplir restricciones de seguridad en todo Java, lo que incluye applets, servlets, JSP e, incluso, el propio Tomcat. Utilizando el Java Security Manager, puede controlar lo que cada aplicación puede y no puede hacer. La tabla 17.2 muestra los tipos de permiso que puede asignar.

Tabla 17.2. Tipos de permisos para Java Security Manager

Tipo de permiso	Significado
java.util.PropertyPermission	Controla el acceso de lectura y de escritura a las propiedades JVM como java.home.
java.lang.RuntimePermission	Controla la utilización de algunos sistemas o de funciones en tiempo de ejecución como exit() y exec().
java.io.FilePermission	Controla permisos a archivos y directorios.
java.net.SocketPermission	Controla la utilización de sockets de red.
java.net.NetPermission	Controla la utilización de conexiones de red multidifusión.
java.lang.reflect.ReflectPermission	Controla la utilización de reflection para realizar una introspección de clase.
java.security.SecurityPermission	Controla el acceso a los métodos de seguridad.
java.security.AllPermission	Permite todo, es lo mismo que no utilizar el Java Security Manager.

Las políticas de seguridad para Tomcat están definidas en el archivo /var/tomcat/conf/tomcat.policy. Este archivo concede permisos utilizando la sintaxis siguiente:

```
grant codeBase code_source {
    permission_type class [name [, action_list]];
};
```

Por ejemplo, el archivo por defecto /var/tomcat/conf/tomcat.policy concede los permisos siguientes:

```
// Ejemplo política webapp
// Por defecto concedemos acceso de lectura al directorio
// webapp y de escritura a workdir
grant codeBase "file:${tomcat.home}/webapps/examples" {
    permission java.net.SocketPermission "localhost:1024-", "listen";
    permission java.util.PropertyPermission "*", "read";
};
```

Los archivos en \${tomcat.home}/webapps/examples (es decir, /var/tomcat/webapps/examples) tienen concedido permiso para utilizar sockets de red para escuchar un host local utilizando el puerto 1024 o puertos superiores y, permitiendo únicamente el acceso de lectura a todas las propiedades JVM. Si quiere que una aplicación llamada /var/tomcat/webapps/your\_app\_name conecte con el servidor Lightweight Directory Access Protocol (LDAP) utilizando el puerto TCP 389, la concesión de permiso que necesita es:

```
grant codeBase "file:${tomcat.home}/webapps/ app_name" {
    permission java.net.SocketPermission "localhost:389",
    "connect";
    permission java.util.PropertyPermission "*", "read";
};
```

Por defecto, Java Security Manager está desactivado para Tomcat. Debe activarlo del siguiente modo:

1. En el archivo /var/tomcat/conf/server.xml debe encontrar lo siguiente:

```
<!-- Uncomment out if you have JDK1.2 and want to use policy

<ContextInterceptor
className="org.apache.tomcat.context.PolicyInterceptor" />

-->
```

Esta asignación por defecto desactiva el Security Manager, de modo que debe eliminar los comentarios para tener lo siguiente:

```
<ContextInterceptor
className="org.apache.tomcat.context.PolicyInterceptor" />
```

2. Reinicie Tomcat utilizando la opción -security. Por ejemplo, /usr/bin/tomcat -security reinicia Tomcat con Java Security Manager.

El JVM lanzará un AccessControlException o un SecurityException cuando el Java Security Manager detecta una violación de la política de seguridad.

# Configurar Apache para Servlets y JSP

Cuando se inicia Tomcat, se crea un archivo de configuración llamado /var/tomcat/conf/mod\_jk.conf-auto. Necesita que Apache cargue este archivo para interaccionar con Tomcat. Para hacerlo, modifique httpd.conf para añadir la línea siguiente:

```
Include /var/tomcat/conf/mod_jk.conf-auto
```

El archivo mod\_jk.conf generado automáticamente, tiene un problema. Le da instrucciones a Apache para que cargue el módulo mod\_jk.so desde un subdirectorio llamado libexec bajo el directorio raíz del servidor (señalando a la directiva ServerRoot en httpd.conf). La realidad es que este directorio no existe y mod\_jk.so está instalado en el directorio /usr/lib/apache. Por lo que tiene dos opciones:

- En lugar de dirigirlo al archivo /var/tomcat/conf/mod\_jk.conf-auto utilizando la directiva `Include`, puede copiar este archivo con otro nombre (por ejemplo, mod\_jk.conf) y utilizar:

```
Include /var/tomcat/conf/mod_jk.conf
```

Entonces puede modificar el archivo mod\_jk.conf para tener la línea `LoadModule jk_module libexec/mod_jk.so` cambiada a `LoadModule jk_module /usr/lib/apache/mod_jk.so`.

- O, puede crear simplemente un directorio llamado libexec dentro del directorio raíz de su servidor Apache y situar mod\_jk.so en él. En este caso, debe iniciar Apache después de iniciar Tomcat porque /var/tomcat/conf/mod\_jk.conf-auto es generado por Tomcat cada vez que se inicia.

Supongo que ha elegido la primera opción, que se gestiona con más facilidad. La primera línea de configuración es:

```
LoadModule jk_module /usr/lib/apache/mod_jk.so
```

La directiva `LoadModule` carga el módulo DSO mod\_jk.so que le permite a Apache interaccionar con Tomcat.

```
JkWorkersFile /var/tomcat/conf/workers.properties
```

Esta línea determina que la directiva `JkWorkersFile` se dirija a /var/tomcat/conf/workers.properties, que proporciona mod\_jk con la información necesaria para conectar con instancias tomcat, que también se conocen con el nombre de workers. Deje esta directiva aparte.

```
JkLogFile /var/tomcat/logs/mod_jk.log
```

Aquí, la ruta del archivo de registro está definida utilizando la directiva `JkLogFile`.

```
JkLogLevel error
```

La línea anterior asigna el nivel de registro para los errores registrados en los archivos de registro. Los niveles posibles de registro son debug, warn, error y emerg, pero warn debería ser su elección por defecto.

```
JkMount /*.jsp ajp12
JkMount /servlet/* ajp12
```

Las dos líneas mostradas asignan los prefijos URL `/*.jsp` y `/servlet/*` al worker Tomcat llamado `asp12`. Esto significa que cualquier solicitud URL que tiene `/<anything>.jps` (por ejemplo, `/foo.jsp`, `/bar.jsp`) o `/servlet/<anything>` (por ejemplo, `/servlet/foo`, `/servlet/bar`) se asignarán al worker `asp12`. Se recomienda que cambie estas dos líneas para que tengamos el worker `asp13` (más rápido), tal y como se muestra a continuación:

```
JkMount /*.jsp ajp13
JkMount /servlet/* ajp13
```

Las directivas mostradas aquí son típicas directivas de Apache:

```
Alias /examples "/var/tomcat/webapps/examples"
<Directory "/var/tomcat/webapps/examples">
    Options Indexes FollowSymLinks
</Directory>
```

Se crea un alias llamado `/examples`, que está dirigido a un directorio físico llamado `/var/tomcat/webapps/examples`. El contenedor de directorios permite realizar una lista de directorios y de enlaces simbólicos en este directorio.

Las siguientes dos directivas asignan `/examples/servlet/*` y `/examples/*` al worker `asp12` de Tomcat:

```
JkMount /examples/servlet/* ajp12
JkMount /examples/*.jsp ajp12
```

Puede cambiarlas a `asp13` si quiere. No es importante porque son simples ejemplos.

El siguiente grupo de directivas desactiva el permiso a los directorios `WEB-INF` y `META-INFO` bajo el árbol de ejemplos. Estos dos directorios no deberían ser navegables, por lo que la medida apropiada es la siguiente:

```
<Location "/examples/WEB-INF/">
    AllowOverride None
    deny from all
</Location>
```

```
<Location "/examples/META-INF/">
    AllowOverride None
    deny from all
</Location>
```

El siguiente grupo de directivas crea exactamente la misma configuración que se creó para /examples.

```
Alias /admin "/var/tomcat/webapps/admin"
<Directory "/var/tomcat/webapps/admin">
    Options Indexes FollowSymLinks
</Directory>

JkMount /admin/servlet/* ajp12
JkMount /admin/*.jsp ajp12

<Location "/admin/WEB-INF/">
    AllowOverride None
    deny from all
</Location>

<Location "/admin/META-INF/">
    AllowOverride None
    deny from all
</Location>
```

Puede cambiar el worker para /admin/servlet/\* y /admin/\*.jsp a ajp13, para sacar partido de este nuevo y veloz protocolo. Finalmente, se repite la misma configuración para /test:

```
Alias /test "/var/tomcat/webapps/test"
<Directory "/var/tomcat/webapps/test">
    Options Indexes FollowSymLinks
</Directory>

JkMount /test/servlet/* ajp12
JkMount /test/*.jsp ajp12

<Location "/test/WEB-INF/">
    AllowOverride None
    deny from all
</Location>

<Location "/test/META-INF/">
    AllowOverride None
    deny from all
</Location>
```

Cuando termine de modificar la configuración anterior y tenga guardado el archivo `httpd.conf`, reinicie el servidor Apache. Puede acceder a los ejemplos de servlet y JSP utilizando `http://your_web_server/examples/`. Tam-

bien puede acceder a la herramienta admin utilizando `http://your_web_server/admin/`. Sin embargo, si quiere ver información sobre contextos (es decir, directorio virtual) utilizando la herramienta admin basada en Web, debe crear un usuario llamado admin en el archivo `/var/tomcat/conf/tomcat-users.xml`. Este archivo es:

```
<tomcat-users>
  <user name="tomcat" password="tomcat" roles="tomcat" />
  <user name="role1"  password="tomcat" roles="role1"  />
  <user name="both"   password="tomcat" roles="tomcat,role1" />
</tomcat-users>
```

En el ejemplo siguiente, he añadido un usuario llamado admin con la contraseña mypwd.

```
<tomcat-users>
  <user name="tomcat" password="tomcat" roles="tomcat" />
  <user name="role1"  password="tomcat" roles="role1"  />
  <user name="both"   password="tomcat" roles="tomcat,role1" />
  <user name="admin"  password="mypwd" roles="admin" />
</tomcat-users>
```

Con este usuario, puede ver los contextos mediante la interfaz Web si activa `trusted = true` para el contexto admin como se muestra a continuación:

```
<Context path="/admin"
  docBase="webapps/admin"
  crossContext="true"
  debug="0"
  reloadable="true"
  trusted="true" >
</Context>
```

La configuración por defecto del contexto admin, tiene el atributo trusted con el valor false. Si le asigna el valor true para poder crear, eliminar o ver contextos mediante la interfaz Web, debe reiniciar Tomcat para que los cambios tengan efecto. Además observe que nunca debe dejar el atributo trusted con el valor true después de usarlo. Dejando esta asignación en trusted, tendrá su servidor abierto a grandes riesgos de seguridad. Por ejemplo, cuando el contexto admin es de confianza (asignando `trusted="true"`) puede crear un contexto mediante `http://your_server_hostname/admin` que expone a la Web su sistema completo de archivos.

Para aprovecharse de esto, el intruso necesita que un sistema Tomcat con contexto admin, esté asignado a trusted y que el archivo XML de texto `/var/tomcat/conf/tomcat-users.xml` tenga un nombre de usuario y una contraseña admin. El mejor modo de enfrentarse a todo esto es eliminar completamente el contexto admin, o al menos asegurarse de que lo ha asignado para que no sea de confianza (es decir, `trusted="false"`).

# Trabajar con Tomcat

Una vez que tiene Tomcat configurado puede empezar a trabajar con él. En esta sección le mostraré cómo desactivar el servicio Web por defecto (en el puerto 80) de Tomcat, lo que implica que sirve páginas estáticas, ya que Apache funciona mejor sirviendo páginas estáticas. Entonces veremos cómo iniciar y parar el servidor Tomcat, y finalmente discutiré un script de shell que se puede utilizar para arrancar el proceso Tomcat.

## Desactivar el servicio HTTP por defecto de Tomcat

Por defecto, Tomcat sirve las solicitudes HTTP en el puerto 8080. Sin embargo, cuando tiene Apache y Tomcat integrados utilizando el módulo mod\_jk.so, no es necesario que Tomcat sirva solicitudes HTTP en el puerto 8080. Puede cambiarlo en la siguiente sección del archivo /var/tomcat/conf/server.xml:

```
<!-- HTTP normal -->
<Connector
  className="org.apache.tomcat.service.PoolTcpConnector">
    <Parameter name="handler"
      value="org.apache.tomcat.service.http.
HttpConnectionHandler"/>
    <Parameter name="port"
      value="8080"/>
</Connector>
```

**NOTA: Debe asegurarse de que Apache no está escuchando en los puertos 8007 o 8080. Compruebe las directivas Listen y Port en httpd.conf para estar seguro.**

Puede desactivar este servicio eliminando la configuración anterior o comentándola, como se hace a continuación:

```
<!-- *** DISABLED Normal HTTP on Port 8080
<Connector
  className="org.apache.tomcat.service.PoolTcpConnector">
    <Parameter name="handler"
      value="org.apache.tomcat.service.http.
HttpConnectionHandler"/>
    <Parameter name="port"
      value="8080"/>
</Connector>
*** DISABLED Normal HTTP Service on Port 8080 -->
```

# Iniciar y parar Tomcat

Antes de iniciar o parar Tomcat, debe asegurarse de que las variables de entorno como \$JAVA\_HOME y \$TOMCAT\_HOME están asignadas, y que la variable \$PATH tiene la ruta \$JAVA\_HOME/bin en ella. Puede utilizar el comando echo para comprobar si estas variables están ya asignadas en su shell. Si ha seguido las secciones anteriores de instalación, debería tener estas variables asignadas en el archivo ~/.bashrc.

Si tiene estas variables de entorno asignadas, puede ejecutar el comando /usr/bin/tomcat start o el comando tomcatctl start para iniciar Tomcat. Puede utilizar el comando /usr/bin/tomcat stop para pararlo. También puede ejecutar el comando /usr/bin/tomcat run para ejecutarlo en primer plano. Sólo se recomienda ejecutarlo en el primer plano para solucionar problemas cuando necesita ver mensajes de error de Tomcat en la consola.

## Iniciar Tomcat con un empaquetador de scripts de shell

Es muy útil crear en entorno necesario en un script de shell y ejecutar estos comandos utilizando un empaquetador de scripts de shell llamado /usr/bin/tomcatctl, que se muestra en el listado 17.2. Esto ahorra una gran cantidad de mecanografía y ayuda a evitar faltas de ortografía.

Listado 17.2. /usr/bin/tomcatctl

```
#!/bin/sh

# Cambie el número de versión abajo si
# JDK no es 1.3.0_02
#
VERSION=1.3.0_02

export JAVA_HOME=/usr/java/jdk$VERSION
export PATH=${PATH}: ${JAVA_HOME}/bin
export TOMCAT_HOME=/var/tomcat

/usr/bin/tomcat $1

# fin
```

Cuando tiene almacenado este script en /usr/bin y ha activado el permiso de ejecución ejecutando el comando chmod 755 /usr/bin/tomcatctl, puede iniciar o parar Tomcat sin preocuparse por las variables de entorno. Por ejemplo, para iniciar Tomcat puede ejecutar el comando tomcatctl start; para parar ejecute el comando tomcatctl stop; y para ejecutarlo en el primer plano, ejecute el comando tomcatctl run.

El paquete binario RPM también instala el script `/etc/rc.d/init.d/tomcat`. Este script se enlaza simbólicamente al nivel de ejecución apropiado, de modo que Tomcat automáticamente inicia y para cada vez que reinicia el sistema. Sin embargo, el valor por defecto para `JAVA_HOME` y para la ruta de JDK en `/etc/rc.d/init.d/tomcat` puede no ser apropiado para todo el mundo. Los valores por defecto asignados en este script para estas dos variables se muestran a continuación:

```
export PATH=$PATH:/opt/IBMJava2-13/bin:/opt/IBMJava2-13/jre/bin  
export JAVA_HOME=/opt/IBMJava2-13
```

Estas parecen ser las asignaciones del propio desarrollo RPM. Si siguió mis instrucciones anteriores para bajar el JDK desde el sitio Web de Sun Microsystems, su ruta de instalación para el JDK será distinta. De modo que asegúrese de modificar estas dos asignaciones en el script `/etc/rc.d/init.d/tomcat`. Por ejemplo, los valores correctos son:

```
export PATH=$PATH:/usr/java/jdk1.3.0_02/bin:/usr/java/jdk1.3.0_02/jre/bin  
export JAVA_HOME=/usr/java/jdk1.3.0_02
```

## Ejecutar servlets de Java

En esta sección vamos a ver cómo podemos preparar Tomcat para que ejecute los ejemplos de servlets que están contenidos en él, y cómo puede ejecutar sus propios servlets o páginas JSP que ha desarrollado o bajado desde Internet.

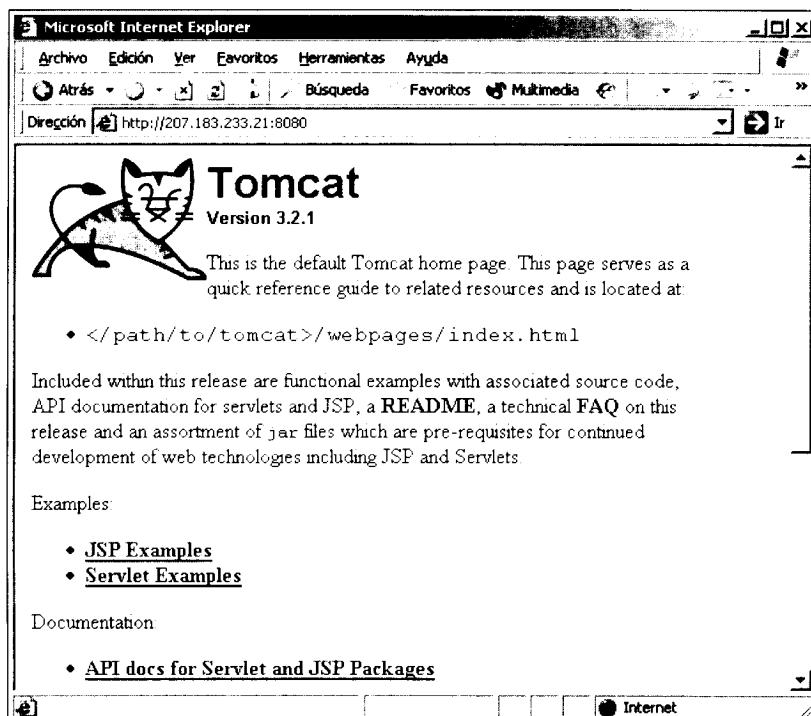
**ADVERTENCIA:** Nunca ejecute Tomcat como raíz. Tiene que crear un nuevo usuario y un nuevo grupo llamado `tomcat` y cambiar los permisos a los archivos y a los directorios para `/var/tomcat` utilizando el comando `chmod -R 755 /var/tomcat`.

### Ejecutar un ejemplo de servlets de Java mediante Tomcat

Para ejecutar servlets de Java, no necesita Apache. Puede utilizar Tomcat, que es un entorno de Java en tiempo de ejecución de código fuente abierto llamado contenedor, para servlets de Java. Tomcat se puede instalar en paralelo a su servidor Web, y puede servir solicitudes servlet estáticas y dinámicas. Sin embargo, Tomcat es mucho más versátil cuando está configurado para servir solicitudes estáticas. No es un simple reemplazo del servidor Web Apache. Si quiere utilizar Tomcat, se encuentra en la sección adecuada.

Por defecto, cuando se inicia Tomcat sirve solicitudes servlet desde el puerto 8080. Cuando ha iniciado Tomcat, debería ser capaz de acceder a los servlets y a las páginas JSP de ejemplo, utilizando `http://localhost:8080` si está

ejecutando el servidor Web en el servidor Tomcat. O, puede acceder a ellos desde otra máquina utilizando `http://your_web_server_hostname:8080/`. Cuando accede por defecto al servidor Tomcat, la página es parecida a la que se muestra en la figura 17.1.



**Figura 17.1.** La página por defecto del servidor Tomcat

Haga clic en el enlace JSP Examples y verá la página mostrada en la figura 17.2. Existen páginas de ejemplo de JSP que puede utilizar para probar su instalación. Por ejemplo, haga clic en el enlace Execute para obtener la página JSP Date. Debería ver una página parecida a la que se muestra en la figura 17.3. El código JSP para esta página JSP está almacenado en `/var/tomcat/webapps/examples/jsp/dates/date.jsp`, que se muestra en el listado 17.2.

#### **Listado 17.2. date.jsp**

```
<html>
<!--
    Copyright (c) 1999 The Apache Software Foundation. Todos los
    Derechos reservados.
-->

<%@ page session="false"%>

<body bgcolor="white">
```

```

<jsp:useBean id='clock' scope='page' class='dates.JspCalendar'
type="dates.JspCalendar" />

<font size=4>
<ul>
<li>    Day of month: is <jsp:getProperty name="clock"
property="dayOfMonth"/>
<li>    Year: is <jsp:getProperty name="clock"
property="year"/>
<li>    Month: is <jsp:getProperty name="clock"
property="month"/>
<li>    Time: is <jsp:getProperty name="clock"
property="time"/>
<li>    Date: is <jsp:getProperty name="clock"
property="date"/>
<li>    Day: is <jsp:getProperty name="clock" property="day"/>
<li>    Day Of Year: is <jsp:getProperty name="clock"
property="dayOfYear"/>
<li>    Week Of Year: is <jsp:getProperty name="clock"
property="weekOfYear"/>
<li>    era: is <jsp:getProperty name="clock" property="era"/>
<li>    DST Offset: is <jsp:getProperty name="clock"
property="DSTOffset"/>
<li>    Zone Offset: is <jsp:getProperty name="clock"
property="zoneOffset"/>
</ul>
</font>

</body>
</html>

```

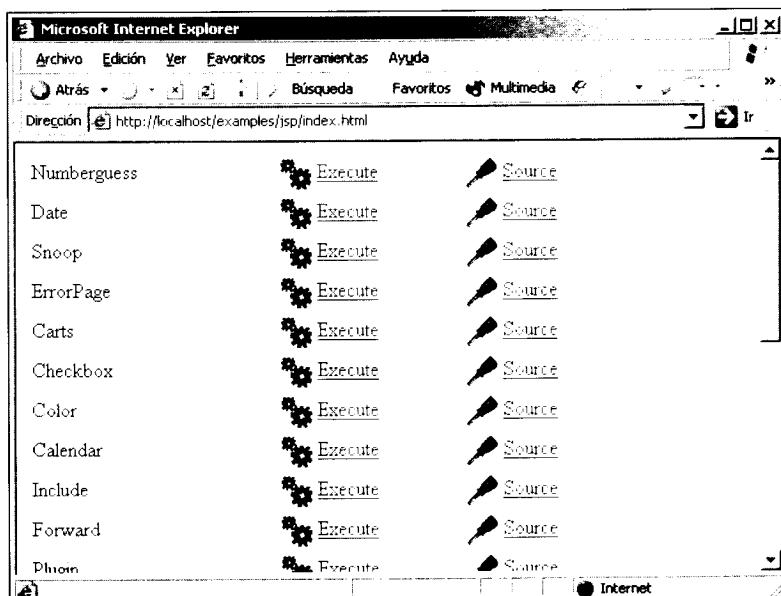


Figura 17.2. Ejemplo de un listado JSP

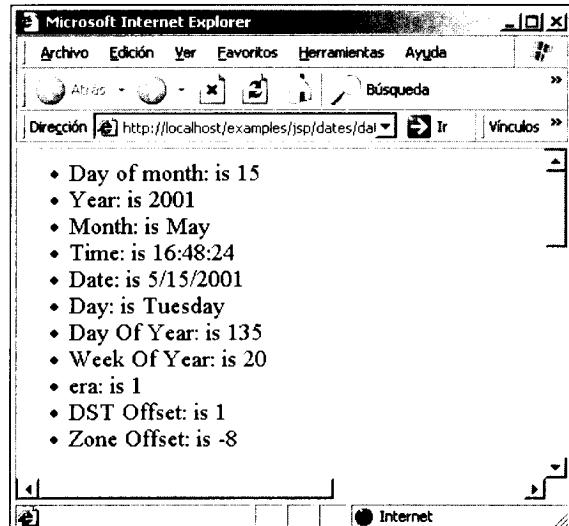


Figura 17.3. Salida del archivo /examples/jsp/dates/date.jsp

Puede ver que el código JSP está embebido en HTML utilizando las etiquetas `<jsp:getProperty name="name" property="property_name""/`.

Si vuelve al inicio de la página y hace clic en el enlace Servlet Examples verá una página como la que se muestra en la figura 17.4.

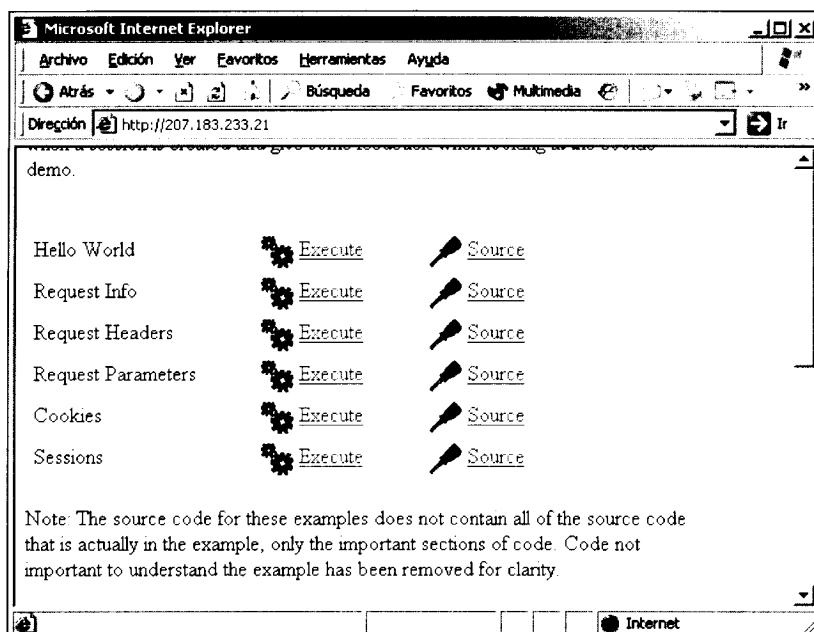


Figura 17.4. Lista de ejemplos de servlet

Haga clic en el enlace Execute para el servlet Hello World y verá la página de la figura 17.5.

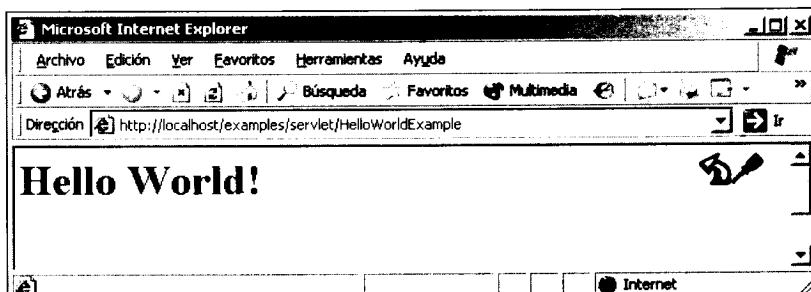


Figura 17.5. Salida de /examples/servlet/HelloWorldExample

Vuelva a la página anterior utilizando el botón de atrás del navegador Web y haga clic en el enlace Source de Hello World para ver el código fuente simplificado para este servlet. Éste se muestra en la figura 17.6.

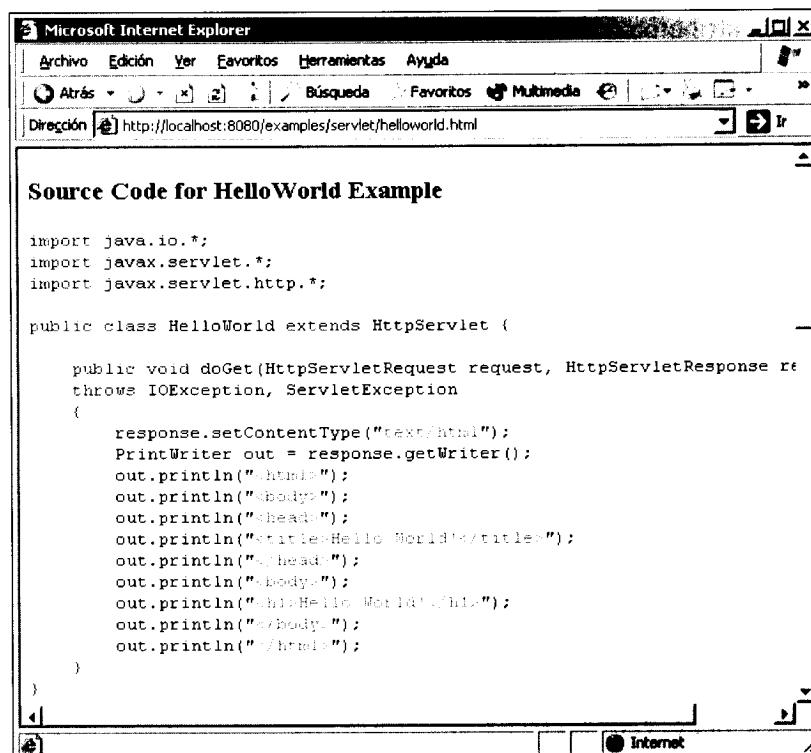


Figura 17.6. Código fuente simplificado de /examples/servlet/HelloWorldExample

Como puede ver, se trata de un servlet bastante sencillo, simplemente imprime un manojo de etiquetas HTML para crear una salida.

## Ejecutar sus propios servlets o JSP

Esta sección le enseña a configurar sus propios servlets para ejecutarlos mediante mod\_jk. Imagine que su servlet se llama myapp. A continuación tiene lo que necesita para que funcione con Apache y con Tomcat:

1. Tiene que crear la siguiente configuración en /var/tomcat/conf/server.xml:

```
<Context path="/myapp"
         docBase="/var/tomcat/webapps/myapps"
         debug="0"
         reloadable="true" >
</Context>
```

El atributo `path` determina la ruta en el URL que indicará su servlet llamado `myapp`. El atributo `docBase` determina la ruta en la que se encuentra el servlet. Ésta puede ser una ruta absoluta o relativa al Tomcat Context Manager (que por defecto es `$TOMCAT_HOME/webapps`). El atributo `debug` determina el nivel de depuración de mensajes de registro, y el atributo `reloadable` determina si Tomcat volverá a cargar un servlet automáticamente cuando cambie.

2. Pare e inicie Tomcat utilizando el comando `/usr/bin/tomcat stop`; `/usr/bin/tomcat start`. Esto generará un nuevo archivo `/var/tomcat/conf/mod_jk.conf-auto`, que tendrá este segmento de configuración:

```
Alias /myapp "/var/tomcat/webapps/myapp"

<Directory "/var/tomcat/webapps/myapp">
    Options Indexes FollowSymLinks
</Directory>

JkMount /myapp/* ajp12
JkMount /myapp/*.jsp ajp12

<Location "/myapp/WEB-INF/">
    AllowOverride None
    deny from all
</Location>

<Location "/myapp/META-INF/">
    AllowOverride None
    deny from all
</Location>
```

3. Tiene que crear los directorios `/var/tomcat/webapps/myapp`, `/var/tomcat/webapps/myapp/WEB-INF`, `/var/tomcat/webapps/myapp/WEB-INF/classes` y `/var/tomcat/webapps/myapp/META-INF`.

4. Almacene el archivo de configuración `web.xml` para el servlet `myapp` en el directorio `/var/tomcat/webapps/myapps/WEB-INF`.
5. Copie las clases del servlet para `myapps` en el directorio `/var/tomcat/webapps/myapps/WEB-INF/classes`.
6. Asegúrese de que el usuario Tomcat (es decir, el ID del usuario utilizado para ejecutar Tomcat) puede acceder al nuevo directorio `/var/tomcat/webapps/myapps` (incluyendo todos los directorios y archivos).

¡Ya está! Debería ser capaz de acceder al servlet utilizando `http://your_web_server/myapps`.



# **Parte IV**

# **Asegurar su sitio Web**



# 18 Seguridad Web

---

## En este capítulo

1. Entendemos el concepto de seguridad Web.
2. Conocemos los puntos de control de seguridad.
3. Elegimos una política de seguridad administrativa.
4. Reconocemos y minimizamos los riesgos de seguridad en CGI y en SSI.
5. Protegemos su contenido Web.

Los administradores novatos no son lo suficientemente conscientes de la necesidad de seguridad. Muchos administradores no piensan en la seguridad hasta que no tiene lugar un problema en ese sentido, y entonces es demasiado tarde. Este capítulo proporciona una idea de los riesgos ocultos que convierten el tema de seguridad en algo realmente importante.

## Entender el concepto de seguridad Web

Desde el momento en el que establece y ejecuta su servidor Web y le hace disponible para el resto del mundo, está abriendo una ventana para que otros se

introduzcan en su red. En realidad esta era su intención, ya que la mayor parte de la gente utilizará de forma adecuada la información disponible en su sitio Web, pero algunas personas se dedicarán a buscar agujeros en esta ventana de modo que puedan obtener información a la que en principio no tenían acceso. Algunos de ellos son gamberros que quieren crear situaciones embarazosas, y otros son ladrones de información. En cualquier caso, si alguien tiene éxito y encuentra esos agujeros, puede encontrar su sitio Web mutilado con material obsceno, o podría incluso perder datos confidenciales. Algunas veces los ataques no afectan directamente a su sitio Web. Los infiltrados pueden utilizar su servidor y otros recursos para obtener acceso a otra red, situándole a usted en un riesgo legal. Es evidente que no deseamos encontrarnos en ninguna de estas situaciones.

Para evitar este tipo de riesgos y de vergüenzas, este capítulo le muestra cómo puede asegurar sus sitios Web. Voy a suponer que los siguientes requisitos de seguridad son aplicables a su sitio Web:

- Mantener la integridad de la información que publica en la Web.
- Prevenir la utilización del host de su servidor Web como punto de entrada en la red de su organización (que podrían convertirse en brechas de confidencialidad, integridad o disponibilidad de recursos de información).
- Prevenir la utilización del host de su servidor Web como área de intrusión en otras redes (que podría dar lugar a que su organización fuese responsable de daños a terceros).

La mayor parte de los incidentes de seguridad ocurren porque la información confidencial es vulnerable debido a una configuración de software indebida. El software puede ser el propio servidor Web, o aplicaciones (como programas CGI, Server-Side Includes y aplicaciones Server API) que se ejecutan en el servidor Web. Esto puede dar lugar a la revelación inadvertida de información confidencial. Se pueden lanzar ataques a partir de este momento con la intención de aprovechar la información confidencial. Aunque la configuración de su servidor esté cerrada y las aplicaciones que se ejecutan en el servidor sean lo suficientemente seguras, sigue sin estar totalmente seguro. Los ataques pueden enfocarse al servidor Web mediante otras rutas. Su servidor Web podría verse comprometido por aplicaciones que no tengan ninguna relación, por un error del sistema operativo, o por una mala arquitectura de red. Para estar seguro, tiene que tener en cuenta todos y cada uno de los detalles. Esto puede resultar una tarea desmoralizante; pero recuerde que es imposible tener un entorno totalmente seguro. No existe una solución infalible para todas las cuestiones de seguridad, porque no todas son conocidas. Su meta tiene que ser mejorar su seguridad lo máximo posible.

## Los puntos de control

El primer paso en la protección de su servidor Web contra intrusos es entender e identificar el riesgo implicado. No hace mucho tiempo, los sitios Web únicamente

mente servían páginas HTML estáticas, que eran menos propensas a los riesgos de seguridad. El único modo en el que un intruso podía piratear en ese tipo de sitios Web era forzando el servidor con un acceso ilegal. Esto se llevaba a cabo normalmente utilizando una contraseña ambigua o engañando a otro software (servidor) demonio. Sin embargo, la mayoría de los sitios Web no sirven páginas HTML estáticas; sirven contenido dinámico, normalmente personalizado, para proporcionar una grata experiencia al usuario. Muchos sitios Web están ligados a aplicaciones para proporcionar servicios de calidad al cliente o para realizar actividades de comercio electrónico. Aquí es donde los riesgos empiezan a aumentar. Las aplicaciones Web son el corazón de los problemas de seguridad en la Web. La mayoría de los sitios Web que han sido atacados por vándalos, no tuvieron problemas con el software del servidor Web. Fueron pirateados debido a uno o más agujeros en la aplicación o en el script que ejecutaba el servidor Web.

Hay varios puntos de control que tiene que revisar para mejorar la seguridad de su servidor Web público. La figura 18.1 muestra el diagrama general de puntos de control. Este diagrama identifica los puntos de control de seguridad que se deben examinar cuidadosamente antes de considerar segura una red conectada con Internet.

Hay tres puntos principales de control:

- **Su red:** este es el punto de control que tiene que ser más estricto, ya que se trata de un punto de entrada principal. Su red conecta con Internet mediante un router, un sistema firewall o un servidor gateway. Por lo tanto, su red es un punto de control principal. Su primera preocupación debería ser asegurar su red.

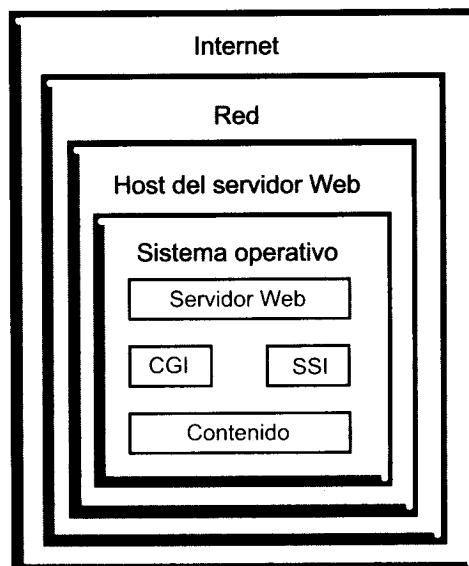


Figura 18.1. Diagrama general de los puntos de control de seguridad

- **Su sistema operativo:** el sistema operativo que utiliza es muy importante. Si ejecuta Apache en una versión de un sistema operativo (Linux, Windows o Solaris) que sea conocida por sus riesgos de seguridad, entonces se puede atacar a su servidor. Es necesario ejecutar un sistema operativo actualizado.
- **El software de su servidor Web:** este es su tercer punto de control. Tiene que ejecutar una versión de Apache que no tenga riesgos de seguridad.

Veremos estos tres puntos de control en las secciones siguientes.

## Punto de control 1: su red

El primer punto de control que tiene que considerar es su red, y cómo está conectada a Internet. La cuestión principal es dónde situar su servidor Web. La idea general es aislarlo de su red interna, mantenerlo lo más lejos posible de ojos fisgones. Definitivamente tiene que considerar instalar un firewall en su router Internet-LAN. Si el firewall no es una opción, puede considerar la opción de un servidor proxy. Un servidor proxy toma una solicitud de la red y la manda a otra red sin conectarlas entre sí. Esto podría ser una buena solución porque, además, proporciona una excelente facilidad de registro.

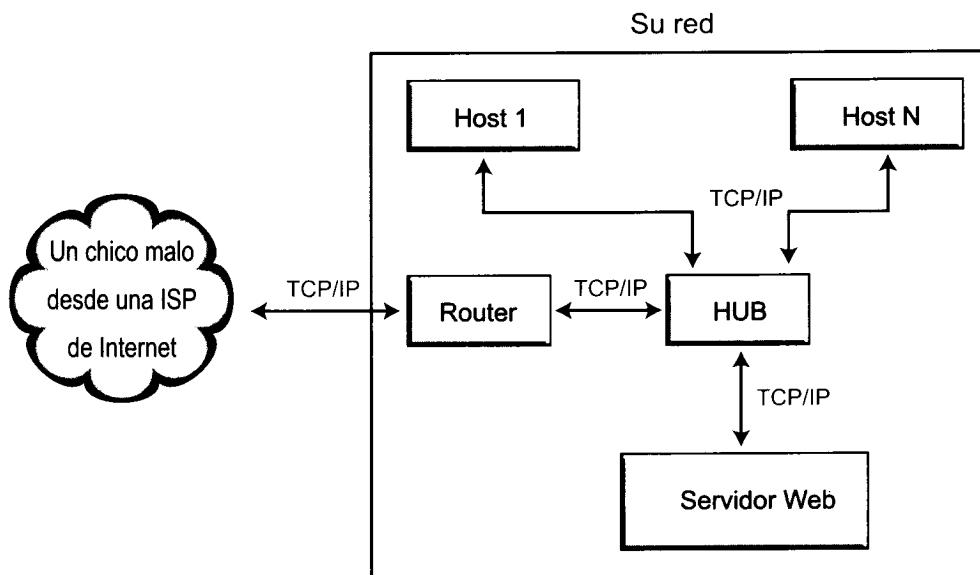
Como puede ver, debería preocuparle más su red interna que el propio servidor Web. Como las redes introducen servidores Web accesibles a Internet dentro de la red, se pueden forzar aprovechando un agujero de seguridad en el software del servidor Web, o un agujero en una aplicación (como un programa CGI) que se esté ejecutando en el servidor Web. Las posibilidades de que esto ocurra se reducen eligiendo una arquitectura que aísle la LAN del servidor Web.

Si tiene una configuración de red como la que se muestra en la figura 18.2, su red está virtualmente desprotegida.

Para facilitar las cosas, voy a suponer que únicamente tiene una conexión a Internet. En esta red, todo el tráfico de Internet se puede dirigir hacia y desde su red. En otras palabras, cualquier persona desde Internet puede llegar a un host de su red y al contrario. Esto también significa que un pirata de Internet puede encontrar un punto débil en uno de los hosts de su red. Por supuesto, no todos estos hosts son servidores con servicios permitidos para otros usuarios; pero, incluso una máquina Windows con compartición de archivos permitida, es un potencial objetivo. Como toda la red utiliza TCP/IP, una simple opción de compartición de archivos en una máquina Windows puede dar lugar a una entrada. Esta red prácticamente está invitando al criminal a robar.

Si tiene esta configuración, compruebe la documentación de su router para determinar las restricciones permitidas basadas en IP. Recomiendo evitar este tipo de configuración.

Remítase a la sección "Elegir una configuración de seguridad" de este capítulo para ver algunas de las configuraciones de red que puede utilizar.



**Figura 18.2.** Una red sencilla con una conexión a Internet

## Punto de control 2: el sistema operativo

Apache se ejecuta en prácticamente todas las plataformas Unix, y se está empezando a ejecutar en Windows 2000, Windows 95, e incluso en el nuevo sistema operativo de Macintosh, Mac OS X. Pero sólo porque Apache se ejecute en prácticamente cualquier plataforma no significa que deba utilizar cualquier sistema operativo para ejecutar Apache. Elegir el sistema operativo (OS) puede ser un factor muy importante en la seguridad del sistema. Los sistemas operativos Unix y basados en Unix más conocidos son Mac OS X, BSDI, FreeBSD, SunOS, Solaris, HP-UX, Digital Unix y Linux. Los administradores de servidores de Internet aprueban estos sistemas operativos. Aunque la calidad y la fiabilidad entre estos sistemas operativos puede variar, todos ellos están diseñados para utilizarlos como sistemas operativos de servidores.

Si no le gusta Unix, investigue la plataforma de servidores Windows 2000. Windows 2000 es un fuerte competidor de Unix; sin embargo, Windows 2000 es nueva en el juego de Internet. La mayoría de los sistemas Unix y los sistemas de tipo Unix llevan el tiempo suficientemente como para haber sufrido numerosos pirateos y ataques. Estos sistemas sobreviven, aunque nunca serán totalmente seguros.

**TRUCO:** Una vez que ha elegido un sistema operativo, tendría que desactivar cualquier característica extra que no tenga intención de utilizar.

**en la máquina de su servidor Web, en el que se ejecuta ese sistema operativo (OS). Por ejemplo, si no necesita los servicios de correo SMTP/POP o los servicios FTP, desactívelos o elimínelos completamente de su sistema. Si no necesita alguna de las utilidades del sistema, eliminelas. Asegúrese de que su servidor Web es sólo un servidor Web y que sólo tiene los servicios que necesita.**

Puede leer sobre las vulnerabilidades que existen en su plataforma operativa, navegando en [www.cert.org](http://www.cert.org). CERT (Computer Emergency Response Team) que trabaja con la comunidad de Internet para facilitar conocimientos y respuestas a las cuestiones de seguridad. La organización también dirige investigaciones enfocadas a la mejora de la seguridad de los sistemas de computadores.

## **Punto de control 3: software del servidor Web**

Obviamente, está utilizando Apache como software de su servidor Web o no estaría leyendo este libro. Asegúrese, sin embargo, de que utiliza la última versión estable del servidor Apache 2 para su sistema. Es una buena idea compilar sus propios binarios Apache en lugar de utilizar los de la distribución de binarios.

Apache es un software disponible gratuitamente, por lo que es importante obtenerlo de una fuente fiable. No descargue binarios de Apache o código fuente de cualquier sitio Web o FTP. Compruebe primero siempre el sitio oficial, [www.apache.org](http://www.apache.org). La fuente de Apache está firmada por PGP (Pretty Good Privacy). Observe, además, que la distribución actual de Apache incluye un archivo llavero PGP que contiene las claves PGP de los desarrolladores de Apache más activos.

Las firmas PGP del grupo Apache están en el archivo KEYS, en el nivel máximo de la distribución de Apache. Si confía en el sitio desde el que está obteniendo su distribución, puede añadir estas claves a su propio archivo llavero. Si no sabe cómo utilizar PGP en su sistema, puede aprenderlo en [www.pgp.com](http://www.pgp.com).

Cada distribución fuente de Apache tiene una firma PGP y estas firmas se almacenan con la extensión .asc (por ejemplo, la firma para apache\_2.0.tar.gz está almacenada en apache\_2.0.tar.gz.asc). Podemos utilizar todo esto para comprobar si la validez de la distribución ha añadido el contenido del archivo KEYS a su archivo llavero.

La última versión de Apache no tiene ningún error conocido; sin embargo, es capaz de incorporar módulos externos de terceras partes, aumentando, de este modo, los riesgos de seguridad, ya que no tiene por qué confiar en un módulo de una tercera parte. Personas de todo el mundo escriben muchos de los módulos de Apache. Asegúrese de que sólo utiliza los módulos disponibles en el sitio oficial de Apache o registrados en el registro de módulos de Apache localizado en <http://modules.apache.org/>.

**TRUCO:** No utilice nunca módulos en experimentación en sus sitios de producción. De hecho, es una buena idea determinar qué módulos necesita realmente, y configurar y compilar Apache de acuerdo con esta determinación. De este modo tendrá un servidor más pequeño y rápido, y potencialmente más seguro.

Cuando configure Apache, preste mucha atención a las cuestiones de seguridad. La idea principal es desactivar todo lo que no utilice. De este modo, está tomando medidas de seguridad preventivas y probablemente reduciendo los riesgos.

## Elegir una configuración segura

Hay muchas formas de configurar Apache para aumentar la seguridad general de su Web. Antes de elegir una configuración adecuada para usted, tiene que desarrollar una política de seguridad y, entonces, podrá elegir una configuración adecuada que cumpla con los requisitos de esta política. Cuando tratamos el tema de seguridad siempre existe la posibilidad de que pueda exagerar los riesgos y volverse loco o subestimar sus necesidades de seguridad y dejar las cosas demasiado expuestas. En las secciones siguientes se verán unas cuantas configuraciones que se adecuan a distintos entornos.

## Consideraciones de política de seguridad

Una política administrativa de seguridad identifica las prácticas que son esenciales para alcanzar una seguridad robusta de red. Si no tiene una aún, considere añadir alguno de los siguientes puntos a su política administrativa de seguridad.

- **Registrelo todo:** no puedo enfatizar este punto lo suficiente. Los archivos de registro registran información sobre el comportamiento del servidor en respuesta a cada solicitud. El análisis de los registros pueden proporcionar tanto información de negocio (por ejemplo, qué páginas Web son más populares) como información de seguridad. Asegúrese de preparar Apache para que registre tanto los accesos como los errores. Con la ayuda de los archivos de registro puede seguir la pista de quién está accediendo a qué. Introdúzcase en los hábitos de navegación siempre que pueda, si observa algo inusual, preste una atención especial. Su registro de error es probablemente el registro que hay que controlar más de cerca.

Si quiere obtener más información sobre registros, remítase a la sección "Registro y seguridad" de este capítulo y a capítulos anteriores en donde describíamos el registro con Apache.

- **Mantener una copia completa de su sitio o sitios Web:** mantenga una copia completa de su sitio Web en un host más seguro. Si la integridad de la información pública está siempre comprometida, necesita una copia completa desde la cual poder restablecer la integridad. Normalmente, la copia completa se guarda en un host al que el administrador del sistema pueda acceder (y quizás también las personas de su organización responsables de la creación y el mantenimiento del contenido Web). A menudo se guarda en la red interna de la organización.

Para garantizar la seguridad, utilice tecnologías robustas de sumas de comprobaciones criptográficas para generar una suma de comprobación para cada archivo. Mantenga las copias completas de archivos y realice las sumas de comprobación del medio protegido para escritura o el de sólo lectura, almacenados en una localización física segura. Puede utilizar encriptado MD5 para generar sumas de comprobación de encriptado para sus archivos.

- **Administre su sitio desde la consola del host:** debería administrar su sitio Web desde la consola del host. Haciendo esto elimina el tráfico de red entre el servidor Web y la terminal del administrador. Existen, sin embargo, muchas situaciones en las que todo esto no es posible (por ejemplo, organizaciones en las que el administrador no tiene facilitado el acceso al servidor). Cuando tiene que realizar una administración remota, tiene que asegurarse de que utiliza un fuerte esquema de autentificación para los registros en el servidor Web. Si utiliza una herramienta administrativa basada en la Web, asegúrese de que no utiliza autentificación HTTP básica. En otras palabras, no puede tener un formato sin encriptar para las contraseñas que están viajando desde su terminal al servidor Web. Además, debería configurar el servidor Web de modo que aceptase una conexión desde un solo host dentro de red interna.
- **Conocer las aplicaciones CGI de dominio público:** cada vez que utilice un CGI de dominio público, asegúrese de que usted o alguno de su organización (o un consultor externo) entiende el código perfectamente. Nunca obtenga una copia de una aplicación de una fuente desconocida. Busque en los grupos de noticias USENET para ver si alguien ha detectado algún problema en la aplicación que está intentando instalar en su servidor Web. Si es posible, instale la aplicación en un servidor de prueba y pruébelo usted mismo. Mantenga un control sobre sus archivos de registro durante el período de prueba y busque errores y advertencias producidos por las aplicaciones. Si la aplicación utiliza servicios de correo, controle también el registro de correos.
- **Compare contenidos:** los gamberros a menudo sustituyen, modifican y dañan archivos del sistema a los que han accedido. Para obtener acceso ilegal al sistema, suelen modificar programas del sistema, de modo que los

programas funcionan de forma normal en apariencia, pero incluyen puertas deatrás para los intrusos. Los vándalos también modifican archivos de registro para eliminar las pistas de sus actividades. Incluso crean nuevos archivos en su sistema.

Para evitarlo, por lo tanto, es una buena idea comparar los atributos y los contenidos de los archivos y de los directorios con la copia completa. Si ha creado sumas de comprobaciones criptográficas para los archivos, puede comparar estas sumas entre las copias actual y completa, para determinar si existe alguna diferencia. Debería buscar en Tripwire, en [www.tripwire.com](http://www.tripwire.com), para obtener detalles sobre el modo de mantener el contenido a salvo utilizando sumas de comprobación.

- **Utilice MD5 para comprobar la integridad del contenido de los archivos:** el programa MD5 genera un único valor de 128 bits para la digestión del mensaje criptográfico derivado del contenido de un archivo. Este valor se considera una huella digital realmente fiable, que se puede utilizar para comprobar la integridad del contenido del archivo. Sólo con que esté modificado el valor de un solo bit, la suma de comprobación MD5 cambiará para este archivo. Es muy difícil falsificar un archivo de modo que el MD5 produzca el mismo resultado que en el archivo original. Un conjunto de sumas de comprobación MD5 para sistemas, aplicaciones y archivos de datos proporciona un modo conciso de almacenar información para utilizarla en comprobaciones periódicas de integridad de estos archivos. Si hay algún cambio que no se pueda atribuir a actividades autorizadas, debería pensar que su sistema está comprometido y debería tomar las medidas necesarias. En el caso de que los cambios sean razonables, vuelva a realizar una copia completa y a realizar las sumas de comprobación.

## Una configuración de seguridad práctica para Apache

Una configuración de seguridad sensata (es decir, no paranoica) para Apache, utiliza un usuario y un grupo especializados sin privilegios para Apache, una estructura de directorios bien definida de los documentos Web y de los archivos de registro, y permisos de archivo y directorio adecuados que permitan únicamente al servidor Web leer y/o escribir en los archivos y directorios. Esta configuración no permite por defecto el acceso y permite el acceso a los recursos (como a los directorios) sólo bajo reglas explícitas. Los detalles de esta configuración se discuten en las siguientes secciones.

### Utilizar un usuario y un grupo especializado para Apache

Se puede ejecutar Apache como un servicio independiente, o como un servicio demonio `inetd`. Si elige ejecutar Apache como un servicio `inetd`, no tiene que

preocuparse por las directivas User y Group. Si ejecuta Apache como un servicio independiente, tiene que crear un usuario y un grupo especializados para Apache.

No utilice el usuario nobody o el grupo nogroup, sobre todo si su sistema ya los ha definido. Es probable que existan otros servicios u otros sitios en los que su sistema los está utilizando. Esto podría derivar en grandes quebraderos de cabeza para el administrador. Tiene que crear un nuevo usuario un nuevo grupo para Apache, y utilizarlos con las directivas mencionadas.

Cuando utiliza un usuario y un grupo especializados para Apache, la administración específica de permisos para su contenido Web resulta más sencilla. Todo lo que necesita, es crear un directorio en el que algunos script CGI puedan escribir datos, tiene que escribir permisos sólo para el usuario Apache.

## Utilice una estructura de directorios segura

En la mayoría de las instalaciones Apache, hay cuatro directorios principales:

1. El directorio específico de ServerRoot, en el que se almacenan la configuración del servidor (subdirectorio conf), los archivos binarios (subdirectorio bin) y otros archivos específicos del servidor.
2. Directorio específico de DocumentRoot en el que se almacena el contenido de su sitio Web, como las páginas HTML, Java Scripts e imágenes.
3. El directorio específico de ScriptAlias en el que se almacenan los scripts CGI.
4. El directorio específico de CustomLog o ErrorLog en el que se almacenan los archivos de acceso y de error. Puede especificar un directorio distinto para cada una de esas directivas pero tiene que mantener un solo archivo de registro para todos los archivos de registro, ya que probablemente se gestione mejor la ejecución de registros.

Le recomiendo que utilice una estructura de directorios en la que cada uno de los directorios principales subdirectorios de otro directorio. ServerRoot debería pertenecer a un directorio al que sólo pudiese acceder el usuario raíz. El directorio DocumentRoot tiene que ser accesible al usuario o usuarios que mantienen su sitio Web y al usuario o al grupo Apache (que se especifica utilizando las directivas User y Group en el archivo httpd.conf). El directorio de script especificado por la directiva ScriptAlias debería ser accesible únicamente a los desarrolladores de script y al usuario o grupo Apache. El directorio especificado por la directiva CustomLog o por la directiva ErrorLog sólo debería ser accesible al usuario raíz.

Ni siquiera el usuario o el grupo Apache deberían tener acceso al directorio de registro. A continuación se muestra un ejemplo de este tipo de estructura de directorios.

```
/  
|  
|  
+---home  
|   |  
|   +---httpd      (ServerRoot)  
|  
+---www  
|   |  
|   +---htdocs     (DocumentRoot)  
|   |  
|   +---cgi-bin    (ScriptAlias)  
|   |  
|   +---logs       (CustomLog and ErrorLog)  
|
```

La estructura de directorios anterior es segura en muchos sentidos. Para entender por qué, observe primero la configuración de Apache en `httpd.conf` para la estructura de directorio anterior.

```
ServerRoot      "/home/httpd"  
DocumentRoot    "/www/htdocs"  
ScriptAlias     /cgi-bin/          "/www/cgi-bin/"  
CustomLog       /www/logs/access.log common  
ErrorLog        /www/logs/error.log
```

Como cada uno de estos directorios principales es independiente (es decir, ninguno es un subdirectorio de otro) están a salvo los unos de los otros. Un error de permiso en un directorio no afectaría a otro directorio. Si no piensa que se trata de un gran logro, cuente el número de años que Linux ha estado implicado con usuarios y entonces lea todas las leyes de Murphy.

Es necesario determinar las asignaciones apropiadas de permisos para la estructura de directorios anterior.

## Permisos de archivos y directorios apropiados

El directorio específico de `ServerRoot` debería ser accesible únicamente al usuario raíz porque sólo la raíz tiene que configurar o ejecutar Apache. `DocumentRoot` debería ser accesible a uno o más usuarios que gestionan contenido en su sitio Web y al usuario Apache (que se ha especificado utilizando la directiva `User`) o al grupo Apache (que se ha especificado utilizando la directiva `Group`). Por ejemplo, si quiere que un usuario llamado `htmlguru` publique contenido en su sitio Web, y que ejecute Apache como usuario `httpd`, puede darle a Apache y a dicho usuario acceso al directorio `DocumentRoot` siguiendo los pasos siguientes:

1. Tiene que crear un nuevo grupo llamado `webteam`:

```
groupadd webteam
```

2. Añada un usuario `htmlguru` al grupo `webteam`:

```
usermod -G webteam htmlguru
```

3. Cambie el propietario de todas las directivas `DocumentRoot` (y de todos los subdirectorios bajo él):

```
chown -R httpd.webteam /www/htdocs
```

Este comando determina que el propietario del directorio sea Apache (es decir, el usuario `httpd`) y el propietario del grupo sea `webteam`, que incluye el usuario `htmlguru`. En otras palabras, tanto Apache como `htmlguru` tendrán acceso al árbol de documentos.

4. Cambie el permiso del directorio `DocumentRoot` (y de todos sus subdirectorios) del siguiente modo:

```
chmod -R 2570 /www/htdocs
```

Este comando garantiza que el usuario Apache puede leer y ejecutar los archivos y los subdirectorios bajo el `DocumentRoot` y que el grupo `webteam` puede leer, escribir y ejecutarlo todo. Además garantiza que cada vez que se crea un nuevo archivo o directorio en el árbol de documentos, el grupo `webteam` podrá acceder a él.

La gran ventaja de este método es que añadir nuevos usuarios a `webteam` es tan fácil como ejecutar el comando siguiente:

```
usermod -G webteam new_username
```

Del mismo modo, si quiere eliminar un usuario existente del grupo `webteam`, simplemente ejecute:

```
usermod -G username [group1,group2,group3,...]
```

donde `group1`, `group2`, `group3`, etc. son grupos (excluido el grupo `webteam`) al que pertenece el usuario.

**TRUCO:** Puede averiguar a qué grupo o grupos pertenece un usuario ejecutando el comando `group username`.

El directorio especificado por `ScriptAlias` debería ser accesible únicamente a los desarrolladores de CGI y al usuario Apache, que se ha especificado utilizando la directiva `User` en `httpd.conf`. Le recomiendo que cree un nuevo grupo llamado `webdev` para el desarrollador o los desarrolladores. Aunque el grupo del desarrollador (como `webdev`) necesite acceso de lectura, escritura y ejecución para el directorio, el usuario Apache sólo necesita acceso de lectura y ejecución. No permita que el usuario Apache escriba archivos en este directorio.

Por ejemplo, imagine que tiene la siguiente directiva ScriptAlias en su httpd.conf:

```
ScriptAlias /cgi-bin/ "/www/cgi-bin/"
```

Si httpd es su usuario Apache y webdev es su grupo de desarrollo, debería asignar permisos para /www/cgi-bin del siguiente modo:

```
chown -R httpd.webdev /www/cgi-bin  
chmod -R 2570 /www/cgi-bin
```

Por otro lado, si quiere permitir a un solo usuario (por ejemplo a cgiguru) desarrollar scripts CGI, puede asignar los permisos de archivo y de directorio del siguiente modo:

```
chown -R cgiguru.httpd /www/cgi-bin  
chmod -R 750 /www/cgi-bin
```

Aquí, el usuario cgiguru es el dueño del usuario, y el grupo (especificado por la directiva Group) utilizado para el servidor Apache es el dueño del grupo del directorio y de sus archivos.

Para terminar, sólo el usuario raíz debería escribir en el directorio de registros utilizado en las directivas CustomLog y ErrorLog. Las asignaciones recomendadas para ese tipo de directorios (por ejemplo /www/logs) son:

```
chown -R root.root /www/logs  
chmod -R 700 /www/logs
```

**ADVERTENCIA:** No permita a nadie (incluidos el usuario o el grupo Apache) leer, escribir o ejecutar en el directorio de registro especificado en las directivas CustomLog y ErrorLog.

Realice una aproximación minimalista de los permisos de acceso a los directorios nuevos que son accesibles mediante la Web. No permita a los visitantes Web ver ninguna lista de directorios. Puede ocultar las listas de directorios utilizando los métodos que se discuten más tarde.

## Archivo index del directorio

Cada vez que un usuario solicita acceso a un directorio mediante la Web, Apache realiza lo siguiente:

1. Comprueba si el directorio es accesible. Si lo es, continúa; en caso contrario, muestra un mensaje de error.
2. Si es accesible, busca el archivo index que se ha especificado utilizando la directiva DirectoryIndex. Por defecto, este archivo es index.html.

Si puede leer este archivo en el directorio solicitado, se muestra el contenido del archivo. En el caso de que no exista ese tipo de archivo, Apache comprueba si puede crear una lista dinámica para el directorio. Si está permitida la creación de una lista dinámica, la crea y muestra el contenido del directorio al usuario.

Como una lista de directorios generada por Apache dinámicamente, puede proporcionar pistas sobre la estructura de su directorio, no debería permitir este tipo de listas. El modo más fácil de evitarlas es crear un archivo con el nombre de archivo especificado en la directiva `DirectoryIndex`. Por ejemplo, si tiene la siguiente directiva `DirectoryIndex`:

```
DirectoryIndex index.html index.htm
```

Apache buscará primero `index.html` en el directorio de la URL, y luego buscará `index.htm` en el caso de no encontrar `index.html`.

Sin embargo, la razón más común por la que muchos sitios Web tienen directorios expuestos es que cuando se crea el directorio nuevo, el creador olvida crear un archivo `index` o descargar un archivo `index` para el caso equivocado, como `INDEX.HTML` o `INDEX.HTM`. Si esto ocurre con cierta frecuencia, puede utilizar un script CGI para redirigir a los usuarios automáticamente a la página de inicio o a una interfaz de una ingeniería de búsqueda interna. Simplemente tiene que modificar la directiva `DirectoryIndex`:

```
DirectoryIndex index.html index.htm /cgi-bin/index.pl
```

A continuación, añada un script CGI como el que se muestra en el listado 18.1 en el directorio especificado por la directiva `ScriptAlias`.

#### Listado 18.1. `index.pl`

```
#!/usr/bin/perl
#
# Propósito: este script se utiliza para redirigir
#           a los usuarios que introducen una URL dirigida a
#           directorios sin página index.html.
#
use CGI qw(:standard);

# Asigna automáticamente la URL redirigida
my $AUTO_REDIRECT_URL = '/';

# Obtiene la ruta de la URL actual
my $curDir = $ENV{REQUEST_URI};

# Si la ruta de la URL actual no es la página de inicio (/)
# entonces
# redirige al usuario a la página de inicio
```

```

if ($curDir ne '/') {
    print redirect($AUTO_REDIRECT_URL);

# Si la página de inicio tampoco tiene la página index,
# no podemos redirigir a la página de inicio (para evitar
redirección
# recurrente), de modo que mostramos un mensaje de error.
} else {
    print header;
    print "HOME PAGE NOT FOUND!";
}

exit 0;

```

Este script se ejecuta cuando Apache falla la búsqueda de los archivos index (index.html o index.htm) en los directorios. El script simplemente redirige un usuario, cuya URL se dirige a un directorio que no tiene archivo index, a la página de inicio del sitio Web.

Cambie /cgi-bin/ desde la ruta de la directiva anterior si utiliza un alias distinto. Si no quiere mostrar ninguna lista de directorios, puede simplemente desactivarlas con la siguiente configuración:

```

<Directory />
    Options -Indexes
</Directory >

```

La directiva Options le dice a Apache que desactive el procesamiento de todos los índices de directorios.

**TRUCO:** Podría ser una buena idea decirle a Apache que no siga enlaces simbólicos, porque este tipo de enlaces pueden exponer, de forma accidental, parte del espacio del disco que no quiere hacer público. Para desactivar el seguimiento de enlaces simbólicos en Apache, fije la directiva Options con el valor ~FollowSymLinks.

## Desactivar el acceso por defecto

Un buen modelo de seguridad implica que no exista acceso por defecto, de modo que debe tomar por costumbre no permitir el acceso por definición. Tiene que permitir el acceso únicamente a directorios determinados. Para implementar que el acceso no sea por defecto, utilice el siguiente segmento de configuración en httpd.conf:

```

<Directory />
    Order deny,allow
    Deny from all
</Directory>

```

Esto desactiva los accesos. Ahora, si tiene que permitir el acceso a un directorio en concreto, utilice el contenedor <Directory . . .> de nuevo, para abrir ese directorio. Por ejemplo, si quiere permitir el acceso a /www/htdocs, añada la configuración siguiente:

```
<Directory "/www/htdocs">
    Order deny,allow
    Allow from all
</Directory>
```

Este método, que consiste en abrir únicamente lo que necesita, es una medida preventiva de seguridad y la recomiendo encarecidamente. Además, no debería permitir a los usuarios cambiar las opciones de configuración de ningún directorio utilizando el archivo de configuración del nivel de directorios (.htaccess) en directorios abiertos al acceso.

## Desactivar invalidación de usuarios

Para desactivar la invalidación de asignaciones de configuración utilizando el archivo de configuración del nivel de directorios (.htaccess) en cualquier directorio, haga lo siguiente:

```
<Directory />
    AllowOverride None
</Directory>
```

Esta configuración desactiva la invalidación de usuarios y, de hecho, aumenta la velocidad de su servidor, porque el servidor no sigue buscando el archivo de control de acceso a directorios (.htaccess) para cada solicitud.

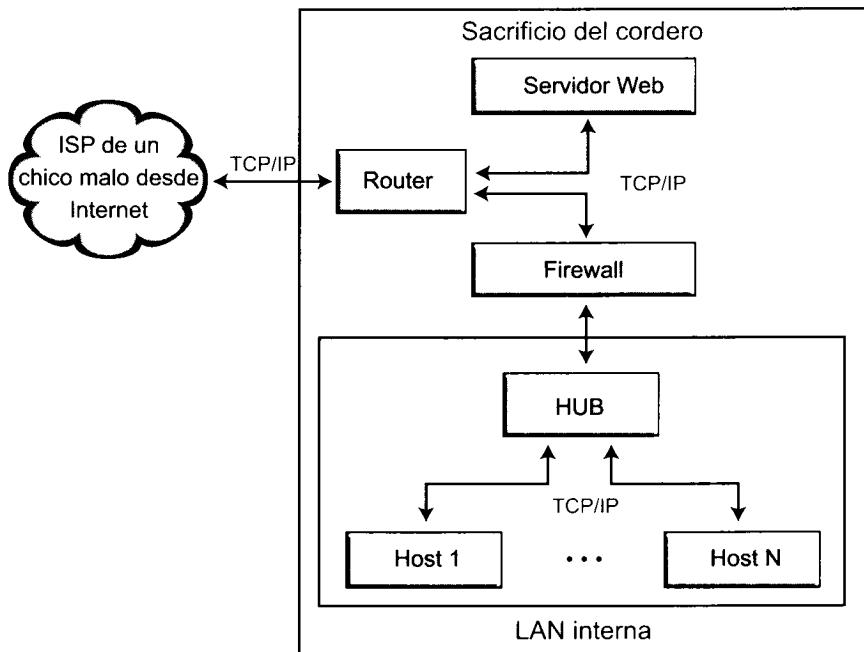
La configuración de Apache que hemos visto, puede ayudarle a crear un servidor Web seguro. Sin embargo, algunos administradores prefieren ir más allá (cerca de la paranoia) en cuestiones de seguridad.

## La configuración "El cordero del sacrificio"

En esta configuración, el servidor Web se mantiene fuera del firewall y de la LAN interna. El hosts de la LAN interna recibe o transmite paquetes de Internet a través de un host del firewall. De este modo, el hosts interno queda protegido de los ataques externos (al menos en teoría). El servidor Web, por otro lado, queda totalmente expuesto al ataque; y este es el motivo por el cual la configuración se denomina "El cordero del sacrificio". La figura 18.3 muestra esta configuración.

En esta configuración:

- El intruso no puede observar ni capturar flujo de tráfico entre los hosts internos. Este tráfico incluye información de autentificación, información sobre los propietarios del negocio, datos personales y este tipo de datos confidenciales.



**Figura 18.3.** Configuración del "cordero del sacrificio"

- El intruso no puede alcanzar host internos, ni obtener información detallada sobre ellos.

La configuración del "cordero del sacrificio" protege contra esos dos tipos de sucesos y mantiene el servidor Web aislado de la red externa y de su tráfico.

**TRUCO:** En esta configuración, es sabio desactivar el enrutamiento de fuente del router. De este modo, no se puede utilizar el host del servidor Web para dirigir paquetes a los hosts de la red interna.

**ADVERTENCIA:** Algunas personas prefieren situar el servidor Web detrás del firewall. En esta configuración, el firewall se convierte en el puente para la LAN interna y para el tráfico Web, y la configuración del firewall se hace más compleja. Considero que esta complejidad puede dar lugar a agujeros de seguridad en el firewall, con el consiguiente fracaso de su propósito.

## La configuración paranoica

Esta configuración es para los administradores Apache que están paranoicos respecto a la seguridad. Es la configuración más restrictiva de todas las configu-

raciones que se discuten en este capítulo. A continuación tenemos los puntos más importantes sobre la configuración paranoica para Apache.

- **No soporta scripts Common Gateway Interface (CGI).** Como se ha mencionado ya, los scripts CGI son la causa típica de la mayoría de los incidentes y por lo tanto no tienen lugar en la configuración paranoica.
- **No soporta SSI.** Al igual que los scripts CGI, las páginas SSI son bastantes problemáticas y por lo tanto tampoco tienen cabida en la configuración paranoica.
- **No permite un sitio Web por usuario.** En el esquema de URL `http://www.domain.com/~username`, se introducen muchas cuestiones de seguridad como por ejemplo que un usuario no tome las precauciones apropiadas para reducir el riesgo de exponer información sobre archivos del sistema al resto del mundo; errores de los usuarios que dan lugar a que áreas de disco del servidor no públicas se vuelvan de acceso público; y a este tipo de situaciones. Por eso los sitios Web para cada usuario no tienen sentido en una configuración paranoica.
- **No permite información de estado mediante la Web.** Apache proporciona un módulo de estado que ofrece valiosa información de estado sobre el servidor mediante la Web. Esta información puede dar pistas a los intrusos si tienen acceso a ella. La configuración paranoica se asegura de no permitir el acceso a este tipo de información careciendo de este módulo.

La configuración paranoica se puede alcanzar utilizando el siguiente comando:

```
./configure --prefix=/home/apache \
--disable-module=include \
--disable-module=cgi \
--disable-module=userdir \
--disable-module=status
```

Una vez que ha ejecutado el comando anterior desde el directorio `src` de la distribución fuente de Apache, puede instalar Apache (en `/home/apache`) con la configuración paranoica.

**TRUCO:** Muchos administradores paranoicos han ejecutado Apache en puertos que no son estándar como 8080 o 9000. Si quiere ejecutar Apache en esos puertos, cambie la directiva `Port` en el `httpd.conf`. Sin embargo, no olvide que los vándalos suelen utilizar programas de escáner para detectar puertos HTTP y que la utilización de este tipo de puertos hace que los usuarios trabajen más duramente porque tienen que teclear el número de puerto (`http://www.domain.com:port/`) al final de cada primera línea de la URL utilizada para entrar en su sitio Web.

# Proteger su contenido Web

El contenido Web es su tesoro y como tal tiene que protegerse del vandalismo y de ataques piratas. Una política de publicación débil o inexistente puede dar lugar a un riesgo en la seguridad de su contenido. En esta sección veremos una política de publicación de contenido con objetivos de seguridad. Además si hay parte de su contenido que no debería indexarse con buscadores Web (por sus robots), entonces puede utilizar las técnicas de control de robots.

## Guías de publicación de contenido

Crear una política de publicación de contenido es una cosa y forzarla es otra. Cuando haya creado su propia política de publicación, discútala con la gente que va a utilizarla. Obtenga sus opiniones sobre cada cuestión y, si es necesario, refine su política para hacerla más útil. Los publicadores de contenido y los desarrolladores de scripts conocen y siguen las siguientes reglas:

- Cada vez que almacene un archivo de contenido, como un archivo HTML, un archivo de imagen, un archivo de sonido, un vídeo clip, y similares, el que publica contenido, debe asegurarse de que el servidor puede leer el archivo (es decir, el nombre de usuario especificado en la directiva `User`). Únicamente la persona que publica puede tener permitido el acceso de escritura en los archivos y en el directorio.
- Cualquier archivo o directorio que no se pueda mostrar directamente en el navegador Web, porque contenga información a la que se accede indirectamente utilizando una aplicación o un script, no debería localizarse en un directorio especificado por `DocumentRoot`. Por ejemplo, si uno de sus scripts necesita acceder a archivos de datos a los que no se debería acceder directamente desde la Web, no guarde el archivo de datos dentro del árbol de documentos. Guarde el archivo fuera del árbol de documentos y obtenga el acceso a su script desde allí, ya que incluso aunque no haya enlaces a estos archivos desde otro contenido visible, podría seguir siendo accesible para otros.
- Cualquier archivo temporal creado por los generadores de contenido dinámico, como aplicaciones CGI, deberían residir en un único subdirectorío en el que tengan acceso de escritura los generadores. Este directorio debe mantenerse fuera del área de contenido para garantizar que un error en la aplicación no destruya de manera equivocada ningún archivo de contenido. En otras palabras, no tenga un directorio en el que pueda escribir el servidor Web dentro de su árbol de documentos. Esto garantiza que un error en el script no escribirá accidentalmente sobre un archivo del árbol de documentos.

- Tiene que mostrar el copyright en el contenido, tiene que haber noticias sobre los derechos de autor visibles y embebidas en las páginas de contenido. Los mensajes embebidos de copyright deberían situarse, si es posible, al comienzo de un documento. Por ejemplo, en un archivo HTML, puede utilizar un par de etiquetas de comentario para embeber el mensaje copyright al comienzo del archivo. Por ejemplo, se puede embeber <!-- Copyright (c) 2001 SuCompañía; Todos los derechos reservados. --> en cada página. Si ha pensado actualizar sus mensajes de copyright a menudo, debería considerar una solución SSI utilizando la directiva #include.
- Si tiene una gran cantidad de contenido en gráficos (imágenes) que quiere proteger de robos de derechos de autor, considere la posibilidad de utilizar tecnología de marcas de agua. Esta tecnología invisible, embebe información en imágenes para proteger el copyright. La idea es que si detecta que un sitio está utilizando su contenido gráfico sin permiso, puede ser capaz de verificar el robo comprobando la información oculta. Si la información se corresponde con la ID de su marca de agua, puede identificar al ladrón y tomar acciones legales. La fuerza de las marcas de agua disponibles actualmente es cuestionable, ya que existen muchos programas que eliminan con facilidad la marca de agua original. Sin embargo, merece la pena investigar esta tecnología si le preocupa su contenido gráfico.

## **Proteger su contenido de robots**

Si quiere proteger realmente su contenido Web tiene que protegerse de robots Web. Todos los buscadores como Yahoo!, AltaVista, Excite e Infoseek utilizan robots automáticos que buscan sitios Web e indexan su contenido. Esto suele ser deseable, pero hay ciertas partes de su sitio Web a las que estos robots no deberían acceder.

Si hay contenido en una sección de su sitio Web que caduca frecuentemente (por ejemplo, a diario), seguro que no quiere que los buscadores lo indexen. ¿Por qué no? Porque cuando un usuario de un buscador encuentra un enlace al contenido antiguo y hace clic sobre él y resulta que este enlace no existe, este usuario se quedará decepcionado respecto a su sitio Web. Este usuario se dirigirá a otro enlace y se olvidará de su sitio web.

Existen otras ocasiones en las que querrá desactivar la indexación de su contenido o parte de él. Estos robots pueden tener efectos intencionados y no intencionados sobre la red y los sitios Web que atraviesan. En ocasiones, los robots de búsqueda están saturados por sitios Web que les están solicitando documentos Web continuamente. Los esfuerzos se dirigen a la creación de estándares de Internet del comportamiento de los robots Web. La versión actual, el protocolo Robot Exclusion, permite a los administradores de los sitios Web situar un archivo robots.txt en sus sitios Web indicando dónde pueden ir los robots. Por ejem-

plo, sería inútil utilizar un robot que está tratando de indexar páginas HTML para un gran archivo de imágenes bitmap. Servir estos archivos al robot supone un uso innecesario de recursos tanto de su servidor como de su robot.

El actual protocolo Robot Exclusion es voluntario, y su etiqueta está evolucionando para los desarrolladores de robots a medida que se obtiene experiencia en el desarrollo de estos robots. El protocolo soporta la mayoría de los buscadores más conocidos.

Cuando un robot visita un sitio llamado `www.somesite.com`, primero comprueba la existencia de la URL `http://www.somesite.com/robots.txt`.

Si esta URL existe, el robot analiza su contenido buscando directivas que instruyan al robot sobre el modo de indexar el sitio. Como administrador Web, puede crear directivas que tengan sentido para su sitio. Observe que sólo puede haber un `/robots.txt` en un sitio. Este archivo contiene registros que serían del siguiente tipo:

```
User-agent: *
Disallow: /cgi-bin/
Disallow: /tmp/
Disallow: /~sam/
```

La primera directiva, `User-agent`, le dice al robot que las siguientes directivas deberían ser consideradas por cualquier robot. Las siguientes tres directivas, `Disallow`, le dicen al robot que no acceda a los directorios mencionados en estas directivas. Observe que tiene que separar la línea `Disallow` de cada prefijo URL que quiera excluir, es decir, no puede utilizar lo siguiente:

```
Disallow: /cgi-bin/ /tmp/ /~sam/
```

Además, no puede tener líneas en blanco en un registro porque las líneas en blanco se utilizan para delimitar varios registros. Las expresiones regulares no se soportan ni en las líneas `User-agent` ni en las líneas `Disallow`. El signo `*` en el campo `User-agent` es un valor especial que significa cualquier robot. No puede tener líneas como las siguientes:

```
Disallow: /tmp/*
o
Disallow: *.gif
```

Todo lo que no está explícitamente desactivado el robot lo considera accesible (a continuación tenemos algunas configuraciones de ejemplo).

## Excluir todos los robots

Para excluir a todos los robots del servidor, utilice la configuración siguiente:

```
User-agent: *
Disallow: /
```

## **Permitir acceso completo a todos los robots**

Para permitir acceso completo a todos los robots, utilice la configuración siguiente:

```
User-agent: *
Disallow:
```

**NOTA:** Puede crear el mismo efecto eliminando el archivo robots.txt.

## **Excluir un solo robot**

Para excluir del acceso a un solo robot llamado WebCrawler, utilice la configuración siguiente:

```
User-agent: WebCrawler
Disallow: /
```

## **Activar un solo robot**

Para permitir el acceso al sitio a un solo robot llamado WebCrawler, utilice la configuración siguiente:

```
User-agent: WebCrawler
Disallow:
User-agent: *
Disallow: /
```

## **Desactivar un solo archivo**

Para desactivar un solo archivo llamado /daily/changes\_toOften.html de la indexación por parte de robots, utilice la configuración siguiente:

```
User-agent: *
Disallow: /daily/changes_toOften.html
```

**NOTA:** Actualmente, no existe la directiva Allow para robots.

## **Registro y seguridad**

Los buenos administradores Web controlan cuidadosamente el registro en sus servidores, lo que proporciona pistas sobre patrones de acceso inusuales. En esta sección veremos el modo de proteger sus archivos de registro de accesos no auto-

rizados y también lo que hay que hacer cuando detecta accesos inusuales en sus archivos de registro.

**ADVERTENCIA:** Los registros son muy útiles, pero si los vándalos pueden modificarlos, se vuelven inútiles. Por lo tanto es necesario que proteja sus archivos. Le recomiendo mantener los archivos de registro en su propia partición, en la que únicamente puede acceder el usuario que tiene que realizar los cambios necesarios.

## CustomLog y ErrorLog

Utilizando las directivas CustomLog y ErrorLog (que se describen en el capítulo 8), Apache le permite registrar solicitudes de acceso que han tenido éxito y solicitudes de acceso que han dado lugar a un error, en archivos de registro separados.

Por ejemplo:

```
CustomLog /logs/access.log common  
ErrorLog /logs/error.log
```

La primera directiva, CustomLog, registra cada solicitud entrante a su sitio Web y la segunda directiva, ErrorLog, sólo registra las solicitudes que generan una condición de error.

El archivo ErrorLog es un buen sitio para llevar un control exhaustivo de los problemas de los que informa su servidor Web. Puede utilizar un programa de análisis de registro como Wusage ([www.boutel.com](http://www.boutel.com)) para analizar de forma rutinaria y controlar sus archivos de registro.

Asegúrese de que únicamente el usuario raíz puede escribir en los directorios especificados por las directivas ServerRoot, CustomLog y ErrorLog. No tiene que conceder permiso de escritura o lectura en los directorios de registro al usuario o al grupo Apache. Permitir a cualquiera, que no sea el usuario raíz, que escriba en los archivos en el directorio de registros puede producir un importante agujero de seguridad. Para garantizar que únicamente el usuario raíz tiene acceso a los archivos de registro en un directorio llamado /logs, siga los pasos siguientes:

1. Cambie el dueño del directorio y de todos sus archivos al usuario raíz y al grupo raíz utilizando el siguiente comando:

```
chown -R root:root /logs
```

2. Cambie el permiso del directorio utilizando el siguiente comando:

```
chmod -R 750 /logs
```

# Qué hacer si observa un acceso inusual en sus archivos de registro

Si observa un acceso inusual, como por ejemplo, alguien tratando de suministrar parámetros inusuales a sus scripts CGI, debería pensar que está teniendo lograr un intento de tipo hostil y debería investigar el asunto inmediatamente. A continuación tiene el procedimiento que puede utilizar:

1. Obtenga la URL completa que está utilizando en la tentativa para engañar al script CGI.
2. Si usted no ha escrito el script, pregúntele al autor del script qué es lo que ocurriría si alguien pasa de esa URL (es decir, cuáles son los parámetros dentro de la URL después de la ?) al script. Si hay alguna razón para preocuparse, comience la investigación, o no la continúe a partir de este punto pero realice una nota de la dirección IP en un archivo POSSIBLE-ATTACKS.txt junto con la URL, y el sello de la hora y la fecha.
3. Si la URL hace que el script haga algo que no debiera, considere la posibilidad de sacar el script de la Web hasta que esté arreglado, de modo que esa URL no pueda suponer una amenaza para el sistema.
4. Utilice el programa del host para detectar el nombre del host de la dirección IP del intruso. Algunas veces el host no podrá encontrar el nombre del host. En este caso, trate de trazar la ruta y de identificar la ISP reconociendo la dirección IP.
5. Realice una búsqueda de dominio whois para la ISP e intente encontrar la información de contacto técnica que se encuentra en las listas del resultado de la búsqueda whois. Debería dirigirse a un sitio Web de registros de dominios para realizar la búsqueda whois en el caso de que no tenga el programa whois instalado. Trate de localizar un registro de dominio apropiado desde el sitio [www.internic.net](http://www.internic.net).
6. Envíe un correo electrónico a la dirección de contacto técnico del ISP relacionado con el incidente y suministre el fragmento de registro para que lo revise. Escriba su correo electrónico educadamente y en tono amistoso. Recuerde que el ISP es su única línea de defensa en este momento. Educadamente solicite una resolución o una respuesta rápida.
7. Si no puede sacar el script de la red porque lo están utilizando en su sitio Web, puede prohibir a los intrusos utilizarlo. Imagine que ejecuta su script bajo el alias de script ext, que está asignado como:

```
ScriptAlias /ext/ "/some/path/to/cgi/scripts/"
```

Cambie la línea anterior a:

```
<Location /ext>
  SetHandler cgi-script
```

```
Options -Indexes +ExecCGI  
AllowOverride None  
Order allow,deny  
Allow from all  
Deny from 192.168.1.100  
</Location>
```

Reemplace 192.168.1.100 con la dirección IP del vándalo. Esta configuración ejecutará su script como siempre para cualquier usuario excepto para el usuario con la dirección IP que se ha metido en la directiva Deny. Sin embargo, si la dirección IP del vándalo utiliza asignación dinámica de direcciones IP para sus clientes, entonces bloquear la dirección IP en cuestión no será efectivo porque el vándalo puede volver con una dirección IP distinta. En ese caso, considere bloquear la red IP completa. Por ejemplo, si la ISP utiliza la red 192.168.1.0, eliminamos el .100 de la directiva Deny de la línea, para bloquear la ISP completa. Esta medida es una medida drástica y podría bloquear a muchos usuarios inocentes de la ISP. Tenga cuidado cuando considere si bloquea o no a toda una ISP.

8. Espere unos cuantos días la respuesta de la ISP. Si no recibe respuesta, trate de contactar con ellos mediante su sitio Web. Si el problema persiste, contacte con su departamento legal para determinar qué acciones legales se pueden tomar para exigir una acción por parte de la ISP.

## Asegurar su implementación CGI

La mayor parte de los expertos Web están de acuerdo en que los mayores factores de riesgo son los scripts CGI o las aplicaciones que se ejecutan en el servidor Web para producir contenido dinámico. Como los scripts CGI son responsables, normalmente, de la creación del contenido dinámico, son los causantes de la mayoría de los daños. Esta sección se centra en los riesgos asociados con los scripts CGI y muestra cómo se pueden reducir estos riesgos.

### Evadir los riesgos CGI con un programa inteligente

De acuerdo con SANS ([www.sans.org](http://www.sans.org)), los riesgos de seguridad relacionados con CGI se encuentran en el segundo puesto del top ten de la lista de aspectos de seguridad en Internet. Sin embargo, CGI no es inherentemente inseguro; son los scripts CGI escritos de forma incorrecta los que dan lugar a los agujeros de seguridad. Realmente, la simplicidad de la especificación CGI hace sencillo que los programadores inexpertos escriban scripts CGI. Estos programadores inexpertos, no se preocupan por los aspectos de seguridad de la red, crean aplicaciones o scripts que funcionan, al tiempo que crean puertas deatrás y agujeros en

el sistema de forma accidental. Las secciones siguientes discuten primero los tipos de riesgos que crean las aplicaciones o los scripts CGI, y después discute las soluciones para estos riesgos.

**NOTA:** Los términos **aplicaciones CGI** y **scripts CGI** se utilizan indistintamente.

## Filtración de información

Los vándalos pueden engañar a muchos scripts CGI para que filtren información disponible en el servidor. Esta filtración ayuda a los vándalos a introducirse en el sistema. Cuanta más información tenga un vándalo de su sistema, mejor se introduce en el sistema. Por ejemplo, si se utiliza esta URL para mostrar la página /doc/article1.html utilizando el script showpage.cgi:

```
http://unsafe-site.com/cgi-bin/showpage.cgi?pg=/doc/  
article1.html
```

un vándalo podría intentar algo como esto:

```
http://unsafe-site.com/cgi-bin/showpage.cgi?pg=/etc/passwd
```

que mostrará el /etc/passwd (archivo password del usuario) a todo el sistema. Esto sólo funciona si el autor de showpage.cgi no protegió el script de este tipo de filtraciones.

## Consumo de los recursos del sistema

Los vándalos pueden utilizar un script CGI mal programado para consumir recursos del sistema hasta el punto de hacer el servidor virtualmente insensible. Por ejemplo, esta URL permite al visitante del sitio ver una lista de anuncios clasificados en un sitio Web:

```
http://unsafe-site.com/cgi-bin/showlist.pl?start=1&stop=15
```

Los parámetros start=1 y stop=15 se utilizan para controlar el número de registros mostrados. Si el script showlist.pl sólo depende de los valores suministrados de start y stop, entonces un vándalo puede editar la URL y suministrar un número alto para el parámetro stop, para conseguir que showlist.pl muestre una lista mayor de lo normal. De este modo un vándalo puede utilizar este conocimiento y sobrecargar al servidor Web con solicitudes que tardan más en procesarse, lo que hará que los usuarios reales esperen y posiblemente se vayan a la competencia.

## Burlarse de los comandos del sistema mediante scripts CGI

En muchos casos, los vándalos suelen tener éxito engañando a un script de correo basado en un formulario HTML, ejecutando un comando del sistema o

repartiendo información confidencial sobre el sistema. Por ejemplo, imagine que tiene un formulario Web, que los usuarios utilizan para registrarse en sus servicios o para proporcionarle respuestas. La mayoría de los sitios con un formulario Web envían una nota de agradecimiento al usuario mediante un correo electrónico. Se utiliza un script CGI para procesar el formulario. El script debería hacer algo parecido a lo siguiente para enviar el correo electrónico:

```
system("/bin/mail -s $subject $emailAddress < $thankYouMsg");
```

La llamada al sistema ejecuta el programa de correo /bin/mail y le suministra el valor de la variable \$subject como Asunto: cabecera, el valor de la variable \$emailAddress como la dirección de correo electrónico del usuario, y redirige el contenido de dicho archivo en la variable \$thankYouMsg. Esto funciona correctamente y nadie debería saber, normalmente, que su aplicación utiliza este tipo de llamadas al sistema. Sin embargo los vándalos siempre están intentando estropear las cosas, de modo que si están interesados en introducirse en su sitio Web podrían intentar introducir valores irregulares en su formulario Web. Por ejemplo, si un vándalo introduce `vandal@emailaddr < /etc/passwd;` como dirección de correo, engañará al script mandando el archivo /etc/passwd a la dirección de correo especificada por el vándalo.

**TRUCO:** Si utiliza la función `system()` en su script CGI, debería utilizar la opción `-T` en la línea `# !/path/to/perl` para activar el modo de comprobación de corrupción de Perl. Debería asignar también el `PATH` (variable de entorno) utilizando la asignación `$ENV{'PATH'} = '/path/to/commands/you/call/via/system'` para aumentar la seguridad.

## Las entradas del usuario realizan determinadas llamadas inseguras al sistema

Hay determinadas llamadas al sistema que son inseguras al utilizar un script CGI. Por ejemplo, en Perl (un lenguaje de programación muy utilizado en CGI) se podrían realizar este tipo de llamadas utilizando las funciones `system()`, `exec()`, `piped open()` y `eval()`. Del mismo modo, en C, las funciones `popen()` y `system()` son daños potenciales de seguridad. Todas estas funciones / comandos suelen invocar un sub shell (como `/bin/sh`) para procesar el comando user. Incluso los scripts shell que utilizan llamadas `system()` o `exec()` están prácticamente abiertos a puertos de entrada para gamberros. Las comillas simples disponibles en los interpretes shell y en Perl para capturar las salidas de los programas como cadenas de texto, también son peligrosas.

Para ilustrar la importancia de una utilización cuidadosa del sistema, considere este segmento de código Perl en apariencia inocente:

```

#!/usr/bin/perl -w
#
# Propósito: demostrar los riesgos asociados a
# un script CGI mal escrito.
#
# Obtiene el nombre del dominio desde la cadena de consulta
# de la variable de entorno.
#
my $domain = $ENV{'QUERY_STRING'};

# Imprime el content type apropiado.
# Como la salida whois es todo texto,
# decidimos utilizar text/plain como content-type.
#
print "Content-type: text/plain\n\n";

# A continuación tenemos la llamada errónea al sistema:
system("/usr/bin/whois $domain");

# Aquí tiene otra llamada errónea al sistema utilizando las
# comillas simples del shell de perl:
#
# my $output = '/usr/bin/whois $domain';
#
# print $output;

exit 0;

```

Este pequeño script de Perl es un puente WHOIS basado en la Web. Si este script se llama `whois.pl`, y se guarda en el directorio `cgi-bin` de un sitio Web llamado `unsafe-site.com`, un usuario puede llamar este script del siguiente modo:

`http://unsafe-site.com/cgi-bin/script.pl?domain=anydomain.com`

El script tomará `anydomain.com` como la variable `$domain` mediante la variable `QUERY_STRING`, lanzará el programa `/usr/bin/whois` con el valor `$domain` como argumento. Esto devuelve el dato desde la base de datos WHOIS que mantiene InterNIC. Esto parece inocente y perfecto pero el script va a causar un desastre. Observe la siguiente línea:

`http://unsafe-site.com/cgi-bin/script.pl?domain=nitec.com;ps`

Realiza una búsqueda WHOIS en un dominio llamado `nitec.com` y además proporciona la salida de la utilidad `ps` de Unix que muestra el estado del proceso. Esto revela información sobre el sistema que no debería estar disponible para la parte solicitante. Utilizando esta técnica, cualquiera puede aprender muchas cosas sobre su sistema. Por ejemplo, reemplazando el comando `ps` por `df` (una utilidad común de Unix que imprime un resumen del espacio del disco) permite a

cualquiera determinar qué particiones tiene y cómo están de llenas. Este agujero de seguridad podría suponer un gran peligro.

¿Cuál es la lección? No hay ninguna introducción de datos segura, y no podemos convertir las llamadas al sistema en objetivos fáciles para el abuso. La siguiente sección nos dice cómo alcanzar estas dos metas.

## El usuario puede modificar datos ocultos en páginas HTML

Como ya sabrá, HTTP es un protocolo sin estado (no guarda información sobre transacciones previas). Muchos desarrolladores Web mantienen información de estado y otros datos importantes en cookies, o en archivos temporales, o utilizando las etiquetas ocultas. Como un usuario puede desactivar las cookies y crear archivos temporales para cada usuario puede resultar engoroso, se utilizan normalmente las etiquetas ocultas. Una etiqueta oculta es algo parecido a lo siguiente:

```
<input type=hidden name="datakey" value="dataValue">
```

Por ejemplo:

```
<input type=hidden name="state" value="CA">
```

Aquí las etiquetas ocultas almacenan "state=CA," que se puede recuperar con la misma aplicación en una llamada posterior. Las etiquetas ocultas suelen encontrarse en aplicaciones Web de varias pantallas. Como el usuario puede cambiar manualmente las etiquetas ocultas, no deberían revelarse. Hay dos modos de protegerse contra la alteración de datos: el desarrollador puede comprobar el dato oculto después de cada uso, o el desarrollador puede utilizar un esquema de seguridad para garantizar que el dato no ha sido alterado por el usuario. El script CGI que se muestra en el listado 18.2, contempla la utilización del algoritmo MD5 de digestión de mensajes RSA Data Security para proteger los datos.

**NOTA:** Los detalles sobre el algoritmo MD5 están definidos en RFC 1321.

**Listado 18.2.** hidden-md5.cgi

```
#!/usr/bin/perl -w
#
# Propósito: este script muestra la utilización de la
#             digestión de mensajes MD5 en una aplicación
#             Web de varias pantallas.
#
# CVS: $Id$
#
#####
```

```

use strict;
use CGI qw(:standard);
use Digest::MD5;

my $query = new CGI;

# Llama a la subrutina handler para procesar los datos del usuario
&handler;

# Termina
exit 0;

sub handler{

#
# Propósito: determinar qué pantalla se muestra
#             y llamar a la subrutina apropiada para
#             desplegarla.
#

# Obtiene el nombre introducido por el usuario (si hay) y la
# dirección de e-mail(si hay) e inicia dos variables
# utilizando los valores del nombre y del e-mail. Nota: la
# primera vez no tendremos valores para estas variables.

my $name    = param('name');
my $email   = param('email');

# Imprime la cabecera Content-Type apropiada y
# además imprime las etiquetas de la página HTML
print header,
      start_html(-title => 'Multi-Screen Web Application Demo');

# Si no tenemos un valor para la variable $name, es que no
# hemos mostrado aún la pantalla, de modo que vamos a
# mostrarla.
if ($name eq ''){

    &screen1;

# Si tenemos un valor para la variable $name pero la
# variable $email está vacía, entonces tenemos que mostrar
# la pantalla 2.
} elsif($email eq '') {

    &screen2($name);

# Tenemos valores para $name y $email, por lo tanto
# mostramos la pantalla 3.
} else {

    &screen3($name, $email);
}

```

```

}

# Imprime las etiquetas HTML de cierre para la página
print end_html;

}

sub screen1{
#
# Propósito: imprimir un formulario HTML que le pida al
# usuario que introduzca su nombre.
#

print h2("Screen 1"),
    hr({-size=>0,-color=>'black'}),
    start_form,
    'Enter name: ',
    textfield(-name => 'name', -size=>30),
    submit(-value => ' Next '),
    end_form;
}

sub screen2{
#
# Propósito: imprimir un formulario HTML que le pida al
# usuario que introduzca su email. También almacena
# el nombre introducido en la pantalla anterior.
#

# Obtiene el nombre
my $name = shift;

# Crea un mensaje de digestión MD5 para el nombre
my $digest = &create_message_digest($name);

# Inserta la digestión como un nuevo parámetro CGI de modo
# que podemos almacenarla utilizando la subrutina
# hidden() de CGI.pm .
param('digest', $digest);

# Ahora imprime la segunda pantalla e inserta los
# valores $name y $digest como datos ocultos.
print h2("Screen 2"),
    hr({-size=>0,-color=>'black'}),
    start_form,
    'Enter email: ',
    textfield(-name => 'email', -size=>30),
    hidden('name'),
    hidden('digest'),
    submit(-value => ' Next '),
    end_form;
}

```

```

sub screen3{
    #
    # Propósito: imprime un mensaje basado en el dato
    # recogido en la pantalla 2.
    # Sin embargo, imprime el mensaje únicamente si
    # el dato introducido no se ha alterado.
    #

    # Obtiene el nombre y la dirección de email
    my ($name, $email) = @_;

    # Obtiene la digestión del valor $name
    my $oldDigest = param('digest');

    # Crea una digestión nueva del valor de la variable $name
    my $newDigest = &create_message_digest($name);

    # Si las digestiones son distintas entonces se ha alterado
    # el dato en la pantalla 2. En este caso, muestra un
    # mensaje de alerta y para el procesamiento.
    if ($oldDigest ne $newDigest){
        return (0, alert('Data altered. Aborted!'));
    }

    # Mientras que el dato sea correcto continúa el proceso.
    print h2("Screen 3"),
        hr({-size=>0,-color=>'black'}),
        p('Your name is ' . b($name) .
            ' and your email address is ' . b($email) . '.'),
        a({-href=>"$ENV{SCRIPT_NAME}"},'Restart');
}

sub create_message_digest{
    #
    # Propósito: crear una digestión de mensajes para el dato dado.
    #             Para conseguir que el vándalo no pueda
    #             reproducir la digestión, esta subrutina
    #             utiliza una clave secreta.
    #

    my $data = shift;

    my $secret = 'ID10t' ; # Cambie esta clave si quiere.

    # Necesitamos la línea siguiente para decirle a Perl
    # que queremos utilizar el módulo Digest::MD5.
    use Digest::MD5;

    # Crea un nuevo objeto MD5
    my $ctx = Digest::MD5->new;

    # Añade datos
    $ctx->add($data);
}

```

```

# Añade una clave secreta
$ctx->add($secret);

# Crea una digestión Base64
my $digest = $ctx->b64digest;

# Devuelve la digestión
return $digest;
}

sub alert{
#
# Propósito: mostrar una caja de dialogo de alerta
#             utilizando Java Script.
#
# Obtiene el mensaje que tenemos que mostrar
my $msg = shift;

# Crea un java script que utiliza la función alert()
# para mostrar un mensaje y entonces
# devuelve al navegador a la pantalla anterior.
print <<JAVASCRIPT;

<script language="JavaScript">
  alert("$msg");
  history.back();
</script>

JAVASCRIPT
}

```

El listado 18.2 es un script CGI de varias pantallas que le pide al usuario que introduzca un nombre en la primera pantalla, después le pide que introduzca una dirección de correo electrónico en la pantalla siguiente, y finalmente imprime un mensaje. Cuando el usuario pasa de una pantalla a otra, el dato de la pantalla anterior pasa a la pantalla siguiente utilizando etiquetas ocultas. Este script funciona tal y como podemos leer en los siguientes párrafos.

Al comienzo del script, se utilizan dos módulos externos, CGI y Digest::MD5. El primero es el módulo CGI más conocido y su función es escribir scripts CGI en Perl de un modo muy sencillo. El segundo módulo nos proporciona una interfaz para el algoritmo MD5.

El primer script crea un objeto CGI llamado \$query. Entonces llama a la subrutina handler() para cada solicitud. La subrutina handler() es el driver del script. Si mira en el interior de la subrutina handler() verá que asigna los campos del nombre completo de usuario y de la dirección de correo electrónico (recogidos de un formulario Web) a \$name y \$email, respectivamente. Entonces imprime un mensaje cabecera Content-Type utilizando el método

`header()` y crea el inicio de un documento HTML utilizando el método `start_html()` encontrado en el módulo CGI.

Si el usuario no ha introducido aún un nombre, esta subrutina muestra la pantalla 1 utilizando la subrutina `screen1()`. Si la variable `$name` no tiene ningún valor, la subrutina `handler()` muestra la pantalla 2 utilizando la subrutina `screen2()`. Cuando `$name` y `$email` tienen valores, se utiliza la subrutina `screen3()` para mostrar la pantalla 3. Finalmente, la subrutina imprime las etiquetas de cierre para la página utilizando el método `end_html()` encontrado en el módulo CGI.

A continuación, observe la subrutina `screen1()`, que es muy sencilla. Simplemente muestra un formulario Web que tiene un campo llamado `name` que el usuario tiene que llenar. Observe que el formulario Web se crea utilizando los métodos `start_form()`, `textfield()`, `submit()` y `end_form()`, que se encuentran en el módulo CGI. No está definida la acción del formulario, lo que significa que se está llamando al script que generó el formulario Web. En otras palabras, cuando se muestra el formulario de la pantalla 1 y un usuario introduce su nombre y hace clic en el botón de enviar, se llama al mismo script con el campo del nombre asignado al dato definido por el usuario.

La subrutina `screen2()` toma la variable global `$name` como un argumento y utiliza la subrutina `create_message_digest()` para crear una digestión MD5 llamada `$digest` para el nombre introducido por el usuario. La digestión se añade al objeto CGI utilizando `param('digest', $digest)`, que se introduce entonces en el formulario Web de la pantalla 2 Web utilizando el método `hidden('digest')` del módulo CGI. El formulario Web de la pantalla 2 Web se crea utilizando los métodos `start_form()`, `textfield()`, `hidden()`, `submit()` y `end_form()` del módulo CGI. El método `hidden('argument')` simplemente crea la etiqueta `<input type="hidden name="argument" value="value_found_in_CGI_object_for_argument">`. El motivo por el que se crea la digestión y se esconde en el formulario Web, junto con el nombre, es porque esto nos permite determinar si se ha alterado el campo del nombre oculto.

Cuando llamamos a la subrutina `screen3()`, toma las variables globales `$name` y `$email` como argumentos. Esto también asigna la variable local `$oldDigest` a la digestión oculta almacenada en el objeto CGI. Entonces crea una nueva digestión llamando a la subrutina `create_message_digest()` utilizando el valor de la variable `$name`. Compara las dos digestiones para determinar si se ha alterado el campo del nombre en la pantalla 2. Recuerde que se introdujo un nombre en la pantalla 1 y que se ocultó utilizando la etiqueta HTML escondida en la pantalla 2. Por eso, existe la posibilidad de que alguien pueda alterar el valor del nombre en tránsito. Si las digestiones no coinciden, se imprime un mensaje de alerta utilizando la subrutina `alert()`, que muestra un mensaje simultáneo en JavaScript. En caso contrario, se imprime el nombre y la dirección de correo electrónico en la pantalla.

La subrutina más interesante es la subrutina `create_message_digest()`. Esta subrutina utiliza cualquier cosa como argumento y utiliza el objeto `Digest::MD5` llamado `$ctx` para añadir dicho dato y la frase de paso secreto (almacenada en `$secret`) utilizando el método `add()` del objeto `Digest::MD5`. Entonces crea una digestión Base64 MD5 utilizando el método `b64digest()`, que se devuelve a la subrutina que se ha llamado. Cuando se ejecuta el script CGI por primera vez, produce una pantalla que le pide al usuario que introduzca su nombre. Cuando el usuario introduce su nombre, pasa a la siguiente pantalla, en la que le piden que introduzca una dirección de correo electrónico. La fuente HTML de esta pantalla se muestra en el listado 18.3.

**Listado 18.3. Fuente HTML para la pantalla 2 de hidden-md5.cgi**

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<HTML>
  <HEAD>
    <TITLE>Multi-Screen Web Application Demo</TITLE>
  </HEAD>

  <BODY>
    <H2>Screen 2</H2>
    <HR SIZE="0" COLOR="black">
    <FORM METHOD="POST"
          ENCTYPE="application/x-www-form-urlencoded">

      Enter email:
      <INPUT TYPE="text"
            NAME="email"
            SIZE=30>

      <INPUT TYPE="hidden"
            NAME="name"
            VALUE="Cynthia">

      <INPUT TYPE="hidden"
            NAME="digest"
            VALUE="IzrSJ1LrsWlYHNfshrKw/A">

      <INPUT TYPE="submit"
            NAME=".submit"
            VALUE=" Next ">

    </FORM>
  </BODY>
</HTML>
```

El dato oculto se almacena utilizando las líneas siguientes:

```
<INPUT TYPE="hidden"
      NAME="name"
```

```
VALUE="Cynthia">  
<INPUT TYPE="hidden"  
NAME="digest"  
VALUE="IzrSJ1LrsWlYHNFshrkW/A">
```

La primera etiqueta oculta almacena name=Cynthia y la segunda almacena digest=IzrSJ1LrsWlYHNFshrkW/A. La segunda parte del dato es la digestión del mismo generado por el nombre en la pantalla 1. Cuando un usuario introduce una dirección de correo electrónico en la segunda pantalla, se muestra la pantalla final.

Sin embargo, antes de que se produzca la pantalla final, se computa una digestión de un mensaje para el nombre en la pantalla 1. La digestión se compara con la digestión que se creó antes, para comprobar que el valor introducido en el campo del nombre de la pantalla 1 no se ha alterado para el campo del nombre de la pantalla 2. Como el algoritmo de digestión MD5 crea la misma digestión del mensaje para un conjunto determinado de datos, cualquier diferencia entre las digestiones producirá indicaciones de advertencia y el script mostrará un mensaje de alerta y no completará el proceso. En otras palabras, si un vándalo decide cambiar el dato almacenado en la pantalla 2 (mostrada en el listado 18.3) y envía el dato para su procesamiento final, la digestión no coincidente permitirá al script detectar el cambio y tomará una acción adecuada. En su mundo real de scripts CGI (escritos en Perl), puede utilizar la subrutina `create_message_digest()` para crear una digestión de mensajes.

**TRUCO:** Puede bajar e instalar la última versión de Digest::MD5 desde CPAN utilizando el comando `perl -MCPAN -e shell` seguido del comando `install Digest::MD5` en el prompt shell CPAN.

## Entradas del usuario seguras

Como puede ver, la mayor parte de los agujeros de seguridad que crean los scripts CGI están causados por entradas inapropiadas. La siguiente sección discute algunos problemas habituales y sus soluciones.

Hay dos aproximaciones para garantizar que la entrada es segura:

- Analizar las entradas buscando caracteres ilegales y reemplazarlos o eliminarlos. Por ejemplo, para el script `whois.pl`, puede añadir la siguiente línea:

```
$domain =~ s/[\\/ ;\\[\\]\\<\\>&\\t]//g;
```

Esto elimina los caracteres meta ilegales. Se trata de una aproximación habitual pero poco aconsejable, que requiere que el programador esté aten-

to a todas las combinaciones de caracteres que pueden dar lugar a un problema. Si el programador no puede predecir la entrada del usuario, existe la posibilidad de que utilice el programa de un modo no previsto por el programador.

- Definir una lista de los caracteres aceptables y reemplazar o eliminar cualquier carácter que no sea aceptable. La lista de los valores de las entradas válidos normalmente es un conjunto de tamaño manejable, predecible y bien definido.

Prefiero la segunda aproximación, ya que no necesita que el programador bloquee todos los caracteres inaceptables, no dejando ningún margen para el error. La aproximación recomendada requiere únicamente que un programador garantice que esos caracteres aceptables estén definidos; de modo que el programador puede despreocuparse en parte de los caracteres que puede intentar introducir un intruso en un intento de evitar las comprobaciones de seguridad.

Basándose en este concepto, el programa `whois.pl` presentado anteriormente podría sanearse para que contenga únicamente aquellos caracteres permitidos; por ejemplo:

```
#!/usr/bin/perl -w
#
# Propósito: Esta es una versión mejor del
# script whois.pl .
#
# Asigna una variable al conjunto de caracteres
# aceptable para los nombres de dominio.
#
my $DOMAIN_CHAR_SET='`-a-zA-Z0-9_';
#
# Obtiene el nombre de dominio de la cadena de consulta
# de la variable de entorno.
#
my $domain = $ENV{'QUERY_STRING'};

#
# Ahora elimina cualquier carácter que no
# pertenezca al conjunto aceptable de caracteres.
#
$domain =~ s/[^\$DOMAIN_CHAR_SET]//g;

#
# Imprime el content type apropiado.
# Como la salida whois output es todo texto, decidimos
# utilizar text/plain como content-type.
#
print "Content-type: text/plain\n\n";

#
# La llamada al sistema:
system("/usr/bin/whois \$domain");

#
# Otra llamada al sistema utilizando las comillas simples de Perl:
```

```

#
# my $output = '/usr/bin/whois $domain';
#
# imprime $output;

exit 0;

```

La variable `$DOMAIN_CHAR_SET` contiene el conjunto aceptable de caracteres, y la variable `$domain` busca todo lo que no pertenezca al conjunto. Los caracteres inaceptables se eliminan.

La mejor forma de abordar las entradas del usuario es establecer reglas para cada entrada (es decir, lo que espera recibir y cómo puede determinar que lo que está recibiendo es aceptable). Por ejemplo, si está esperando una dirección de correo electrónico como entrada (en lugar de analizarla a ciegas buscando meta caracteres shell), utilice una expresión regular como la siguiente, para detectar la validez de la entrada como una dirección posible de correo electrónico:

```

$email = param('email-addr');

if ($email =~ /^[^\w\-.]+\@\w\-.+\$/)
{
    print "Possibly valid address."
}
else {
    print "Invalid email address.";
}

```

Pero sanear las entradas del usuario no es suficiente. Tiene que ser cuidadoso con la forma de invocar programas externos (por ejemplo, hay muchos modos en los que puede invocar programas externos en Perl). Algunos de estos métodos incluyen el uso de las comillas simples de shell para capturar la salida de un programa externo:

```
$list = '/bin/ls -l /etc';
```

Esta línea capture la lista de directorios `/etc`. O, puede abrir una tubería a un programa:

```
open (FP, " | /usr/bin/sort");
```

También puede invocar un programa externo y esperar que vuelva con `system()`:

```
system "/usr/bin/lpr data.dat";
```

O puede invocar un programa externo y que nunca vuelva con `exec()`:

```
exec "/usr/bin/sort < data.dat";
```

Todas estas construcciones son arriesgadas si implican que la entrada del usuario pueda contener meta caracteres shell. Para `system()` y `exec()`, hay caracte-

rística sintáctica algo oscura que le permite llamar programas externos en lugar de hacerlo a través de un shell. Si pasa los argumentos a un programa externo como elementos separados en una lista en lugar de en una gran cadena, Perl no atravesará el shell y los meta caracteres del shell no producirán efectos indeseados. Por ejemplo:

```
system "/usr/bin/sort", "data.dat";
```

Puede sacar partido de esta característica de abrir una tubería sin atravesar el shell. Llamando a la secuencia de caracteres - | se separa una copia de Perl y se abre una tubería a la copia. Entonces, la copia hija inmediatamente separa otro programa utilizando el primer argumento de la llamada a la función exec.

Para leer desde una tubería sin abrir un shell, puede hacer algo parecido con la secuencia - | :

```
open(GREP,"-|") || exec "/usr/bin/grep",$userpattern,$filename;
while (<GREP>) {
    print "match: $_";
}
close GREP;
```

Estas formas de open () son más seguras que el método open () entubado y, por lo tanto, debería utilizarlas siempre que sea posible.

Observe que hay muchas otras características oscuras en Perl que le permiten llamar a programas externos y engañarle sobre su nombre. Esto resulta muy útil para llamar a programas que se comportan de distinto modo dependiendo del nombre con el que se invocan. La sintaxis es:

```
system $real_name "fake_name","argument1","argument2"
```

Un truco que utilizan los vándalos es cambiar la variable de entorno PATH de modo que se dirija al programa que quieren que ejecute su script, en lugar del programa que usted espera que se ejecute. Debería invocar programas utilizando nombres completos de rutas en lugar de depender de la variable de entorno PATH. Es decir, en lugar de este fragmento de código Perl:

```
system("cat /tmp/shopping.cart.txt");
```

utilice este otro:

```
system "/bin/cat", "/tmp/shopping.cart.txt ";
```

Si tiene que depender de la variable PATH, determinela usted mismo al principio de su script CGI, del siguiente modo:

```
$ENV{'PATH'}="bin:/usr/bin:/usr/local/bin";
```

Incluso si no depende de la variable PATH cuando invoca un programa externo, hay una posibilidad de que invoque al programa; por lo tanto, tiene que incluir

la línea anterior al comienzo de su script cada vez que utilice comprobación de contaminaciones. Tiene que ajustar la línea a la lista de directorios que quiera encontrar. Además, en general, no es una buena idea poner el directorio actual (.) en la ruta.

## Empaquetar scripts CGI

El mejor modo de reducir los riesgos relacionados con CGI es no ejecutar ningún script CGI; sin embargo, en los tiempos del contenido dinámico, esta propuesta es muy poco realista. Quizá pueda centralizar todos sus scripts CGI en una localización y controlar escrupulosamente su desarrollo para garantizar que están bien escritos.

En muchos casos, especialmente en los sistemas ISP, todos los usuarios con sitios Web quieren acceso CGI. En esta situación, podría ser una buena idea ejecutar scripts CGI bajo el ID de usuario (UID) de aquel usuario que sea el dueño del script CGI. Por defecto, los scripts CGI que ejecuta Apache utilizan el UID de Apache. Si ejecuta estas aplicaciones utilizando el UID del dueño, el posible daño se limita al acceso que tiene permitido el UID. En otras palabras, un mal script CGI, que ejecuta un UID distinto del UID del servidor Apache, puede dañar únicamente a los archivos del usuario. El usuario responsable del script CGI se volverá más cuidadoso, porque el posible daño afectaría a su contenido únicamente. De una sola vez, ha aumentado la responsabilidad del usuario y ha limitado el área posible de riesgo. Para ejecutar un script CGI utilizando un UID distinto del UID del servidor Apache, necesita un tipo especial de programa llamado wrapper (empaquetador), que le permite ejecutar un script CGI como usuario propietario del archivo, en lugar de como el usuario del servidor Apache. Algunos wrappers CGI realizan otras comprobaciones de seguridad antes de ejecutar los scripts solicitados. Las siguientes secciones discuten dos wrappers CGI.

### suEXEC

Apache tiene una aplicación de soporte llamada suEXEC que proporciona a los usuarios de Apache la posibilidad de ejecutar programas CGI y SSI bajo UID diferentes de la UID de Apache. suEXEC es un programa wrapper setuid, que es llamado cuando se realiza una solicitud HTTP de un programa CGI o SSI, para el que el administrador ha determinado que se ejecute con una UID distinta a la del servidor Apache. Cuando se realiza este tipo de solicitud, Apache proporciona el wrapper suEXEC con el nombre del programa y el UID y el GID. suEXEC ejecuta el programa utilizando dichas UID y GID.

Antes de ejecutar los comandos CGI o SSI, el wrapper suEXEC realiza una serie de pruebas para garantizar que las solicitudes son válidas. Junto con otras cosas, estas pruebas aseguran que el script CGI pertenece al usuario que tiene permiso para ejecutar el wrapper y que únicamente el dueño puede escribir en el directorio CGI o en el script CGI. Si estas comprobaciones de seguridad tienen

éxito, el wrapper suEXEC cambia el UID y el ID de grupo (GID) con el UID y el GID objetivo mediante llamadas `setuid` y `setgid`, respectivamente. La lista de grupos de acceso se inicia con todos los grupos a los que pertenece el usuario. suEXEC limpia el entorno del proceso estableciendo una ejecución segura de PATH (definida durante la configuración), así como atravesando sólo aquellas variables cuyos nombres se encuentren en la lista del entorno seguro (que también se crea durante la configuración). El proceso suEXEC se convierte entonces en el objetivo de los scripts CGI o de los comandos SSI y se ejecuta. Esto parece mucho trabajo, y lo es, pero proporciona un alto coeficiente de seguridad.

## Configurar e instalar suEXEC

Si está interesado en instalar soporte suEXEC en Apache, ejecute el script `configure` (o `config.status`) del siguiente modo:

```
./configure --prefix=/path/to/apache \
            --enable-suexec \
            --suexec-caller=httpd \
            --suexec-userdir=public_html \
            --suexec-uidmin=100 \
            --suexec-gidmin=100 \
            --suexec-safepath="/usr/local/bin:/usr/bin:/bin"
```

A continuación tiene la explicación detallada de esta configuración:

- `--enable-suexec`: permite soporte suEXEC.
- `--suexec-caller=httpd`: cambia `httpd` al UID que utiliza para la directiva `User` en el archivo de configuración de Apache. Este es el único usuario que tendrá el acceso permitido para ejecutar el programa suEXEC.
- `--suexec-userdir=public_html`: define el subdirectorio que se encuentra bajo los directorios locales del usuario, en los que se guardan los ejecutables de suEXEC. Cambie `public_html` a cualquier valor que utilice para la directiva `UserDir`, que determina el directorio raíz de documentos para un usuario del sitio Web.
- `--suexec-uidmin=100`: define el menor UID que puede ejecutar scripts CGI basados en suEXEC. En otras palabras, las UID por debajo de este número no serán capaces de ejecutar scripts CGI o comandos SSI mediante suEXEC. Observe su archivo `/etc/passwd` para asegurarse de que el rango elegido no incluye la cuenta del sistema en la que normalmente las UID son inferiores a 100.
- `--suexec-gidmin=100`: define el menor número de GID que puede ser objetivo de un grupo. En otras palabras, las GID por debajo de este número no serán capaces de ejecutar scripts CGI o comandos SSI mediante suEXEC. Observe su archivo `/etc/group` para asegurarse de que el

rango elegido no incluye grupos de la cuenta del sistema en la que normalmente las GID son inferiores a 100.

- `--suexec-safepath="/usr/local/bin:/usr/bin:/bin"`: define la variable de entorno PATH que ejecuta suEXEC para scripts CGI y comandos SSI.

## Activar y probar suEXEC

Una vez que tiene instalado su wrapper suEXEC y el nuevo ejecutable de Apache en la localización adecuada, reinicie Apache, lo que hará que se escriba un mensaje parecido al siguiente:

```
[notice] suEXEC mechanism enabled (wrapper: /usr/local/sbin/suexec)
```

Esto le informa de que suEXEC está activo. Ahora, compruebe la funcionalidad de suEXEC. Añada las líneas siguientes al archivo httpd.conf:

```
UserDir public_html  
AddHandler cgi-script .pl
```

La primera directiva (`UserDir`) determina que la ruta del documento de un sitio Web de un usuario sea `~username/public_html`, en la que `username` puede ser cualquier usuario del sistema. La segunda directiva asocia el manejador `cgi-script` con archivos `.pl`. Esto se lleva a cabo de modo que los scripts de Perl con extensiones `.pl` puedan ejecutarse como scripts CGI. Para esta prueba, necesitará una cuenta de usuario. En este ejemplo, he utilizado el host `wormhole.nitec.com` y un usuario llamado `kabir`. Copie el script que se muestra en el listado 18.4 en un archivo `test.pl` y situelo en un directorio `public_html` de usuario. En mi caso, he colocado el archivo en el directorio `~kabir/public_html`.

**Listado 18.4.** Un script CGI para comprobar el soporte suEXEC

```
#!/usr/bin/perl  
#  
# Asegúrese de que la línea anterior está dirigida a la  
# localización correcta. Algunas personas guardan perl en  
# /usr/local/bin.  
  
my ($key,$value);  
print "Content-type: text/html\n\n";  
print "<h1>Test of suEXEC<h1>";  
  
foreach $key (sort keys %ENV){  
    $value = $ENV{$key};  
    print "$key = $value <br>";  
}  
exit 0;
```

Para acceder al script mediante un navegador Web, he solicitado la siguiente URL: <http://wormhole.nitec.com/~kabir/test.pl>.

Se ejecuta un script CGI sólo después de que pase todas las comprobaciones de seguridad realizadas por suEXEC. suEXEC también registra las solicitudes en su archivo de registro. La entrada de registro para mi solicitud es la siguiente:

```
[2001-03-29 16:00:22]: uid: (kabir/kabir) gid: (kabir/kabir)
cmd: test.pl
```

Si está realmente interesado en saber si el script se está ejecutando bajo la UID del usuario, inserte un comando `sleep` (como el `sleep(10);`) dentro del bucle `foreach`, lo que hará más lenta la ejecución y le permitirá ejecutar comandos como `top` o `ps` en la consola de su servidor Web para saber el UID del proceso que ejecuta `test.pl`. También puede cambiar el dueño del script utilizando el comando `chown`; trate de acceder al script mediante su navegador Web después de cambiar el dueño, y verá un mensaje de error que registra suEXEC. Por ejemplo, cuando cambio el dueño del script `test.pl` en el directorio `~kabir/public_html`:

```
chown root test.pl
```

Obtengo un error del servidor, y el archivo de registro muestra la siguiente línea:

```
[2001-03-29 16:00:22]: uid/gid (500/500) mismatch with
directory (500/500) or program (0/500)
```

Aquí, el dueño del programa es UID 0, y el grupo sigue siendo kabir (500), de modo que suEXEC rechaza su ejecución, lo que significa que suEXEC está haciendo lo que se supone que tiene que hacer.

Para garantizar que suEXEC ejecutará `test.pl` en otros directorios, he creado un directorio `cgi-bin` en `~kabir/public_html` y he colocado `test.cgi` en ese directorio. Una vez determinado que el grupo y el usuario dueños del nuevo directorio y del nuevo archivo son el ID del usuario `kabir` y el ID del grupo `kabir`, accedí al script utilizando el comando siguiente:

```
http://wormhole.nitec.com/~kabir/cgi-bin/test.pl
```

Si tiene hosts virtuales y quiere ejecutar los programas CGI y/o los comandos SSI utilizando suEXEC, debe utilizar las directivas `User` y `Group` dentro del contenedor `<VirtualHost . . .>`. Asigne estas directivas con las ID de usuario y grupo distintas de aquellas que está utilizando el servidor Apache. Si solo una, o ninguna, de estas directivas es específica para el contenedor `<VirtualHost>`, se supone que hay que utilizar las ID de grupo y usuario del servidor.

Por razones de seguridad y de eficacia, todas sus solicitudes suEXEC deben permanecer dentro de la raíz de documentos de máximo nivel para solicitudes del

host virtual o en la raíz de documentos personal de máximo nivel para las solicitudes userdir. Por ejemplo, si tiene configurados cuatro host virtuales, tiene que estructurar todas las raíces de documentos de los host virtuales fuera de la jerarquía de documentos principal de Apache para sacar partido de suEXEC para los host virtuales.

## CGIWrap

CGIWrap es parecido al programa suEXEC en tanto que permite a los usuarios utilizar scripts CGI sin comprometer la seguridad del servidor Web. Los programas CGI se ejecutan con el permiso del dueño del archivo. Además, CGIWrap realiza varias comprobaciones de seguridad en el script CGI y no se ejecuta si falla alguna de las comprobaciones. Nathan Neulinger escribió CGIWrap; la última versión de CGIWrap está disponible en el sitio FTP principal en `ftp://ftp.cc.umr.edu/pub/cgi/cgiwrap/`. CGIWrap se utiliza mediante una URL en un documento HTML. CGIWrap se configura para ejecutar scripts de usuario que están localizados en el directorio `~/public_html/cgi-bin/`.

### Configurar e instalar CGIWrap

CGIWrap se distribuye como un archivo tar comprimido gzip. Puede descomprimirlo utilizando gzip y extrayéndolo utilizando la utilidad tar.

Ejecute el script `Configure`, que le pide que responda a varias preguntas. La mayoría de estas preguntas se responden por sí mismas. Además observe que hay una característica en este wrapper que es distinta de suEXEC. Le permite crear archivos `allow` y `deny` que se pueden utilizar para restringir el acceso a sus scripts CGI. Ambos archivos tienen el mismo formato, tal y como se muestra a continuación:

```
User ID  
mailto:Username@subnet1/mask1, subnet2/mask2. . .
```

Puede tener un solo nombre de usuario (UID no numérico) o una línea `mailto:ID@subnet/mask` en la que se definen uno o más pares de subred/máscara de red. Por ejemplo, si encontramos la siguiente línea en el archivo `file`,

```
mailto:kabir@192.168.1.0/255.255.255.0
```

los hosts que pertenecen a la red 192.168.1.0, con una máscara de red 255.255.255.0, tienen permiso para ejecutar los scripts CGI del usuario kabir.

Tras ejecutar el script `Configure`, debe ejecutar la utilidad `make` para crear el ejecutable CGIWrap.

### Activar CGIWrap

Para utilizar la aplicación wrapper, copie el ejecutable CGIWrap en el directorio `cgi-bin` del usuario. Este directorio debe coincidir con el directorio que ha

especificado en el proceso de configuración. El modo más sencillo de que funcionen las cosas es guardar el tipo ~username/public\_html/cgi-bin en la estructura de directorios para el directorio de scripts CGI.

Una vez que ha copiado el ejecutable CGIWrap, cambie el propietario y los permisos del siguiente modo:

```
chown root CGIWrap  
chmod 4755 CGIWrap
```

Tiene que crear enlaces de hardware o enlaces simbólicos llamados nph-cgiwrap, nph-cgiwrapd y cgiwrapd para CGIWrap en el directorio cgi-bin:

```
ln [-s] CGIWrap cgiwrapd  
ln [-s] CGIWrap nph-cgiwrap  
ln [-s] CGIWrap nph-cgiwrapd
```

En mi servidor Apache, he especificado únicamente la extensión cgi como script CGI; por lo tanto, he vuelto a nombrar mi ejecutable CGIWrap a cgiwrap.cgi para que funcione. Si tiene restricciones parecidas, podría intentar esta aproximación o confeccionar un enlace.

Ahora, ejecute un script CGI del siguiente modo:

```
http://www.yourdomain.com/cgi-bin/cgiwrap/username/scriptname
```

Para acceder al script CGI de kabir, test.cgi, en el sitio wormhole.nitec.com, por ejemplo, yo utilizaría:

```
http://wormhole.nitec.com/cgi-bin/cgiwrap/kabir/test.cgi
```

Si quiere ver el resultado de la depuración de errores para su CGI, especifique cgiwrapd en lugar de cgiwrap, tal y como se hace en la siguiente URL:

```
http://www.yourdomain.com/cgi-bin/cgiwrapd/username/scriptname
```

Si el script es un script nph-style, debe ejecutarlo utilizando la siguiente URL:

```
http://www.yourdomain.com/cgi-bin/nph-cgiwrap/username/  
scriptname
```

## Ocultar pistas sobre sus scripts CGI

Cuando un vándalo escanea un sitio Web para encontrar posibles agujeros, busca pequeñas cosas que proporcionen pistas sobre el hardware y el software más importantes de su sistema. Por eso, cuantas menos pistas proporcione sobre su sistema, mayor es la posibilidad de que su Web no se convierta en una víctima. Hay varias formas de ocultar algunos detalles importantes que se pueden convertir en pistas.

## Utilice un alias de script que no sea estándar

La utilización del alias cgi-bin es demasiado habitual. Tan pronto como ve una URL con cgi-bin sabe que el sitio ejecuta scripts CGI de alguna clase. Este alias se asigna utilizando la directiva `ScriptAlias` en el archivo `httpd.conf` de Apache. Por ejemplo:

```
ScriptAlias /cgi-bin/ "/path/to/real/cgi/directory/"
```

Pero sólo algunas personas se dan cuenta de que se puede utilizar cualquier cosa para crear un alias como este. Por ejemplo:

```
ScriptAlias /apps/ "/path/to/real/cgi/directory/"
```

Ahora, el `apps` en una URL tiene la misma función que `cgi-bin`. Por eso, si utiliza algo parecido a:

```
ScriptAlias /dcon/ "/path/to/real/cgi/directory/"
```

puede confundir a algunos vándalos debido a la utilización de `dcon`, o cualquier alias no estándar que utilice. Además recuerde que muchos vándalos utilizan programas automáticos para escanear los sitios Web buscando pistas. Un alias de script no estándar como el anterior, es bastante improbable que se incorpore a un programa automático.

## Utilice nombres sin extensión para sus scripts CGI

Hay muchos sitios que muestran el tipo de scripts CGI que ejecutan. Por ejemplo:

```
http://www.domain.com/cgi-bin/show-catalog.pl
```

proporciona dos pistas sobre el sitio. La primera es que indica que el sitio soporta scripts CGI, y la segunda es que el sitio ejecuta scripts Perl como scripts CGI. Si el sitio anterior utilizase:

```
http://www.domain.com/ext/show-catalog
```

sería más difícil determinar algo sobre el sitio. Debe evitar la utilización de extensiones `.pl` o `.cgi` porque estas extensiones proporcionan pistas sobre el sistema. Para cambiar una extensión `.pl`, `.cgi`, y similares, a un nombre sin extensión, simplemente vuelva a nombrar el script. No necesita cambiar o añadir ninguna nueva configuración en Apache para pasar a nombres sin extensión.

Al igual que los scripts CGI, los scripts SSI conllevan algunos riesgos de seguridad. Se discuten más tarde.

## Utilizar escáneres CGI

Los escáneres CGI se utilizan para escanear un servidor Web en busca de vulnerabilidades relacionadas con los scripts CGI. Hay dos tipos de escáneres que son de mi agrado: `cgichk.pl` y Whisker.

## **cgichk.pl**

Este es un sencillo escáner CGI escrito en Perl. Puede bajar la fuente de [www.packetstorm.securify.com](http://www.packetstorm.securify.com). Cuando ejecutamos desde la línea de comandos utilizando el comando perl cgichk.pl, le pedirá que introduzca el nombre del host del servidor Web que quiera escanear y un número de puerto (80 por defecto). También puede decidir registrar los resultados en un archivo.

cgichk.pl primero comprueba la versión del protocolo HTTP que está utilizando el servidor Web. Por ejemplo, la siguiente sesión muestra que cgichk.pl está escaneando un host llamado rhat.nitec.com.

```
CGI scanner [in Perl] v1.1

Host: rhat.nitec.com
HTTP Port [80]:
Log Session?(y/n)y
Log File [rhat.nitec.com.scan]:
Press [enter] to check the httpd version...

HTTP/1.1 200 OK
Date: Tue, 27 Mar 2001 04:50:47 GMT
Server: Apache/2.0.14 (Unix)
Last-Modified: Mon, 26 Mar 2001 20:23:13 GMT
ETag: "1ba42-1000-c65eee40"
Connection: close
Content-Type: text/html; charset=ISO-8859-1
```

Tras detectar la versión del protocolo, cgichk.pl le pedirá que presione la tecla enter para iniciar la comprobación de vulnerabilidades de CGI. La siguiente salida es un ejemplo de un escaneado para encontrar asuntos relacionados con la seguridad en CGI en el servidor Web rhat.nitec.com, que está ejecutando Apache 2.0.

```
Searching for UnIG - backdoor      : Not Found
Searching for THC - backdoor       : Not Found
Searching for phf                  : Not Found
Searching for Count.cgi           : Not Found
Searching for test-cgi            : Not Found
Searching for nph-test-cgi        : Not Found
Searching for nph-publish         : Not Found
Searching for php.cgi              : Not Found
Searching for handler              : Not Found
Searching for webgais              : Not Found
Searching for websendmail          : Not Found
Searching for webdist.cgi          : Not Found
Searching for faxsurvey             : Not Found
Searching for htmlscript            : Not Found
Searching for pfdisplay             : Not Found
Searching for perl.exe              : Not Found
Searching for wwwboard.pl          : Not Found
```

Searching for www-sql : Not Found  
Searching for view-source : Not Found  
Searching for campas : Not Found  
Searching for aglimpse : Not Found  
Searching for glimpse : Not Found  
Searching for man.sh : Not Found  
Searching for AT-admin.cgi : Not Found  
Searching for filemail.pl : Not Found  
Searching for maillist.pl : Not Found  
Searching for jj : Not Found  
Searching for info2www : Not Found  
Searching for files.pl : Not Found  
Searching for finger : Not Found  
Searching for bnbform.cgi : Not Found  
Searching for survey.cgi : Not Found  
Searching for AnyForm2 : Not Found  
Searching for textcounter.pl : Not Found  
Searching for classifields.cgi : Not Found  
Searching for environ.cgi : Not Found  
Searching for wrap : Not Found  
Searching for cgiwrap : Not Found  
Searching for guestbook.cgi : Not Found  
Searching for edit.pl : Not Found  
Searching for perlshop.cgi : Not Found  
Searching for anyboard.cgi : Not Found  
Searching for webbbs.cgi : Found!  
Searching for environ.cgi : Not Found  
Searching for whois\_raw.cgi : Not Found  
Searching for \_vti\_inf.html : Not Found  
Searching for service.pwd : Not Found  
Searching for users.pwd : Not Found  
Searching for authors.pwd : Not Found  
Searching for administrators : Not Found  
Searching for shtml.dll : Not Found  
Searching for shtml.exe : Not Found  
Searching for args.bat : Not Found  
Searching for uploader.exe : Not Found  
Searching for rguest.exe : Not Found  
Searching for wguest.exe : Not Found  
Searching for bdir - samples : Not Found  
Searching for CGImail.exe : Not Found  
Searching for newdsn.exe : Not Found  
Searching for fpcount.exe : Not Found  
Searching for counter.exe : Not Found  
Searching for visadmin.exe : Not Found  
Searching for openfile.cfm : Not Found  
Searching for exprcalc.cfm : Not Found  
Searching for dispopenedfile : Not Found  
Searching for sendmail.cfm : Not Found  
Searching for codebrws.asp : Not Found  
Searching for codebrws.asp 2 : Not Found  
Searching for showcode.asp : Not Found  
Searching for search97.vts : Not Found

```
Searching for carbo.dll      : Not Found
Server may have CGI vulnerabilities.
```

Observe la línea en negrita. El escáner ha encontrado un riesgo potencial. El script `webbbs.cgi` puede ser utilizado por chiquillos que quieren convertirse en piratas para corromper el sistema. Si su escáner identifica uno o más riesgos de seguridad, considere la posibilidad de eliminar esos scripts o de actualizarlos con los ajustes adecuados.

## Whisker

Whisker es un escáner CGI basado en Perl que me gusta mucho. Puede bajar la distribución fuente de [www.filesearch.ru](http://www.filesearch.ru). Una vez que lo ha bajado, extraiga la fuente en un directorio y ejecute el script `whisker.pl` como `whisker.pl -h hostname`. El comando `perl whisker -h rhat.nitec.com` ejecuta el escáner en el servidor Web Apache Web ejecutando dicho host. El resultado se muestra a continuación:

```
= Host: rhat.nitec.com
= Server: Apache/2.0.14 (Unix)

+ 200 OK: HEAD /cgi-bin/webbbs.cgi
+ 200 OK: HEAD /manual/
+ 200 OK: HEAD /temp/
```

El resultado del escáner utiliza códigos de estado HTTP como 200, 303, 403 y 404 para indicar riesgos de seguridad. Por ejemplo, el resultado anterior muestra que hay tres riesgos potenciales (200) en el servidor. Si quiere más información, ejecute el `whisker` con las opciones `-i` y `-v`. Por ejemplo, el comando `perl whisker.pl -h www.domain.com -i -v` se ejecuta en `www.domain.com`. A continuación tiene un ejemplo de una salida de un escáner:

```
= - = - = - = - =
= Host: www.domain.com
- Directory index: /

= Server: Apache/1.3.12 (Unix) mod_oas/5.1/

- www.apache.org
+ 302 Found: GET /scripts/
+ 403 Forbidden: GET /cgi-bin/
+ 200 OK: HEAD /cgi-bin/upload.pl
+ 403 Forbidden: HEAD /~root/
+ 403 Forbidden: HEAD /apps/
+ 200 OK: HEAD /shop/
+ 200 OK: HEAD /store/
```

Observe que hay algunas líneas 200 OK que significan que existe aprovechamiento de recursos; estado 403 que significa que se ha denegado el acceso a un

recurso aprovechable pero que sigue existiendo, esto es bueno y malo al mismo tiempo. Es bueno porque tal y como está configurado el servidor, el recurso aprovechable no se encuentra accesible pero, si en el futuro cambia la configuración, el recurso aprovechable se haría disponible y por eso 403 es al mismo tiempo bueno y malo. La línea 302 indica falsos positivos. Esto ocurre porque muchos servidores están configurados para responder con un mensaje de error personalizado cuando se pierde una URL solicitada, lo que genera un código de estado HTTP 302.

También puede utilizar la opción `-I n` (donde n = de 0 a 9) para activar el modo evasivo y evitar Intrusion Detection System (IDS) en el servidor Web. Si está utilizando cualquier solución IDS, puede probar también la efectividad de su IDS.

Por ejemplo, si su IDS es consciente de `/cgi-bin/phf` (un riesgo CGI conocido), entonces, utilizando `-I 1` intentará engañar a su IDS utilizando codificación de URL de modo que la solicitud `/cgi-bin/phf` se envía a una URL codificada en lugar de utilizar directamente `/cgi-bin/phf` en la solicitud. Del mismo modo, `-I 2` tratará de confundir a IDS utilizando un patrón `/ . /` extra en la URL. Para obtener más detalles, ejecute `whisker` sin ningún argumento.

## Reducir riesgos SSI

Si ejecuta aplicaciones externas utilizando comandos SSI como `exec`, el riesgo en seguridad es prácticamente el mismo que con los scripts CGI. Sin embargo, puede desactivar este comando fácilmente bajo Apache, utilizando la directiva `Options` del siguiente modo:

```
<Directory />
    Options IncludesNOEXEC
</Directory>
```

Esto desactiva `exec` e incluye comandos SSI en todo su espacio Web; sin embargo, puede activar estos comandos cada vez que los necesite definiendo un contendor de directorios con un alcance más limitado, por ejemplo:

```
<Directory />
    Options IncludesNOEXEC
</Directory>

<Directory "/ssi">
    Options +Includes
    SetOutputFilter INCLUDES
</Directory>
```

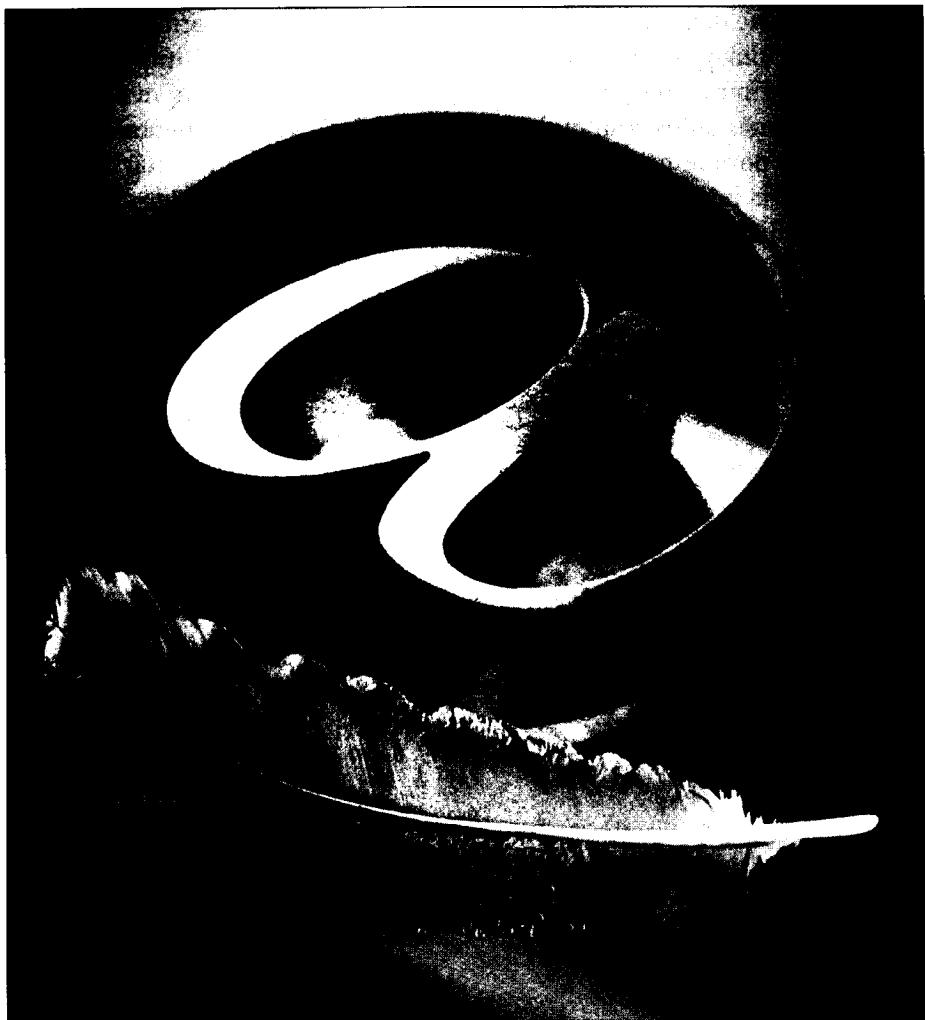
Este segmento de configuración desactiva el comando `exec` en el directorio `/ssi`.

Además tiene que evitar utilizar el comando `printenv`, el cual imprime una lista con todas las variables de entorno y sus valores. Por ejemplo, el comando:

```
<--#printenv -->
```

muestra todas las variables de entorno disponibles para el servidor Web. Mostrando este tipo de información en una página accesible al público le estamos ofreciendo pistas a los gamberros. De modo que tiene que utilizar este comando sólo cuando esté depurando llamadas SSI y nunca en un entorno de producción.

Tiene que tomar una gran cantidad de decisiones (qué permitir y qué no permitir) sobre configuración y política de seguridad para asegurar su servidor Web. Muchas personas se frustrarán después de implementar un conjunto de medidas de seguridad porque no sabrán qué más necesitan para aumentar la seguridad. Una vez que ha implementado un conjunto de medidas como las que se realizan con las solicitudes CGI y SSI, debería centrar sus esfuerzos en el registro.



# 19 Asegurar Apache con SSL

---

## En este capítulo

1. Entendemos cómo funciona SSL.
2. Establecemos OpenSSL.
3. Compilamos e instalamos `mod_ssl` para Apache.
4. Compilamos e instalamos Apache-SSL para Apache.
5. Obtenemos certificados para los servidores.
6. Creamos una autoridad certificada privada.

Hasta hace apenas hace unos años, Internet continuaba siendo lo que fue en sus inicios, una red mundial para científicos e ingenieros. Sin embargo, gracias a la Web, Internet se ha convertido en una red para todos. Parece que todo el mundo y todo las cosas están ahora en Internet. Además se trata de una nueva frontera económica, miles de negocios, grandes y pequeños, se han establecido en sitios de comercio electrónico para abrir sus puertas al mundo. Sin embargo, los clientes son cautelosos, porque saben que no todo es seguro en Internet. Para eliminar la sensación de inseguridad en esta nueva frontera, Netscape Communications (ahor-

ra una sucursal de AOL Time Warner) inventó un protocolo de seguridad, que garantizaba las transacciones entre el navegador Web del cliente y el servidor Web. Netscape denominó a este protocolo Secured Socket Layer (SSL). Rápidamente SSL encontró su sitio en muchas otras aplicaciones de Internet como en el correo electrónico, en el acceso remoto, etc. Como SSL no forma parte de los fundamentos de la infraestructura de seguridad de computadores moderna, es importante saber cómo incorporar SSL en su servidor Apache.

Apache no tiene, por defecto, soporte de SSL. Sin embargo, es posible compilar e instalar un módulo SSL para Apache con el fin de activar capacidades SSL. Hay dos soluciones de código abierto disponibles para Apache: mod\_ssl y Apache-SSL. Estas soluciones utilizan una implementación SSL de código fuente abierto llamada OpenSSL. En este capítulo discutiremos el modo de compilar e instalar OpenSSL y entonces establecer Apache con mod\_ssl o con Apache-SSL.

## Introducción a SSL

Utilizando esquemas asimétricos y simétricos de encriptación (que se describen en la sección "Entender la encriptación"), Netscape desarrolló el protocolo abierto no propietario llamado Secured Socket Layer (SSL) para proporcionar encriptación de datos, autenticación del servidor, integridad de datos y autenticación de clientes para comunicación basada en TCP/IP. La figura 19.1 muestra cómo interacciona SSL con las aplicaciones.

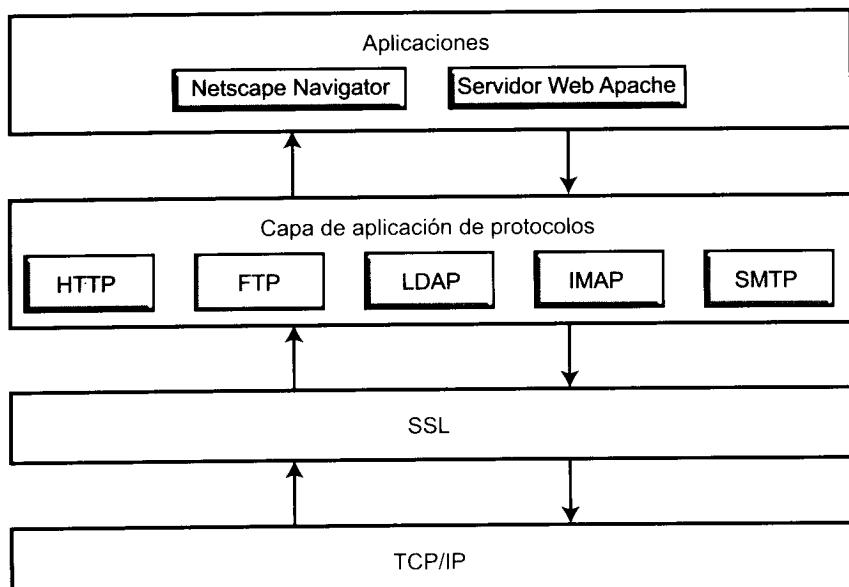


Figura 19.1. Interacciones de SSL con las aplicaciones

El protocolo SSL se ejecuta por encima de TCP/IP y por debajo del máximo nivel de aplicación, es decir, los protocolos de la capa de aplicación como HTTP, FTP, IMAP, y similares. TCP/IP utiliza en parte los protocolos de la capa de aplicación, y en el proceso, activa un servidor que activa SSL para autenticarse a sí mismo en el servidor, permitiendo que ambas máquinas establezcan una conexión de encriptación. La siguiente sección proporciona un resumen del funcionamiento de SSL.

## Cómo funciona SSL

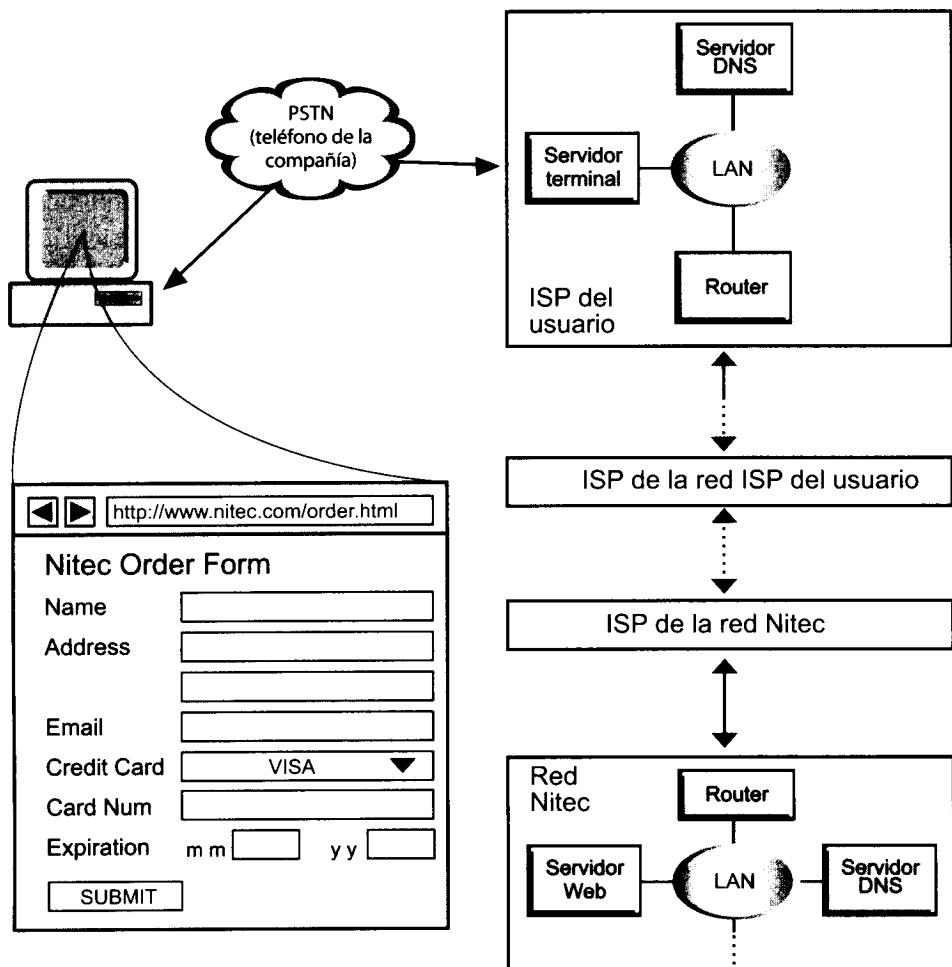
El fundamento de SSL es la encriptación. SSL define cómo y qué tipo de encriptación utilizar para garantizar la seguridad en la comunicación en una red. En las siguientes secciones discutiremos los distintos tipos de encriptación y sus aplicaciones en SSL.

### Entender la encriptación

Cuando los datos viajan de un punto a otro de Internet, atraviesan una gran cantidad de computadoras como routers, gateways, y otros dispositivos de red. Por ejemplo, cuando un visitante del sitio Web [www.nitec.com](http://www.nitec.com) introduce el número de su tarjeta de crédito en un formulario HTML, es muy posible que la información siga un camino parecido al que se muestra en la figura 19.2.

Como puede ver en la figura, los datos deben atravesar varios nodos, de modo que hay una posibilidad de que sean interceptados por alguien. Aunque los paquetes de datos viajan a alta velocidad (normalmente milisegundos), la intercepción sigue siendo posible. Este es el motivo por el que necesitamos un mecanismo seguro para intercambiar datos confidenciales. Esta seguridad se alcanza con la encriptación. Técnicamente hablando, la encriptación es un esquema de codificación matemática que garantiza que únicamente el recipiente propuesto puede acceder a los datos; oculta los datos a los curiosos. Los esquemas de encriptación se utilizan mucho para restringir el acceso a los recursos. Por ejemplo, si se registra en un sistema Unix o Windows 2000/NT, las contraseñas o claves que utilizará se almacenarán en el servidor con formato encriptado. En la mayoría de los sistemas Unix, se encripta una contraseña del usuario y se hace corresponder con la contraseña encriptada almacenada en el archivo /etc/passwd. Si la comparación tiene éxito, el usuario obtiene permiso para acceder al recurso solicitado. Hay dos esquemas de encriptación disponibles:

- **Encriptación simétrica:** este esquema es parecido al de llave y cerradura que utilizamos en la vida diaria. Bloqueamos nuestro coche con una llave, y lo abrimos con esa misma llave. Del mismo modo, en la encriptación simétrica, tenemos una llave o clave que bloquea y abre las utilidades. En la figura 19.3 se muestra un ejemplo de este tipo de esquema.



**Figura 19.2.** Datos viajando desde un punto a otro de Internet

Como se utiliza una sola clave, todas las partes implicadas deben conocer qué es lo que va a hacer esta clave para que este esquema funcione.

- **Encriptación asimétrica:** la encriptación asimétrica funciona de un modo un poco distinto de la encriptación simétrica, tal y como sugiere su nombre. Con este esquema, hay dos claves: la clave pública y la clave privada. La clave extra es la clave pública, por eso este esquema también se conoce con el nombre de encriptación de clave pública. La figura 19.4 muestra un ejemplo del funcionamiento de este esquema.
- Como se muestra en la figura, cuando se encripta un dato con la clave pública, sólo se puede desencriptar con la clave privada, y al contrario. A diferencia de la encriptación simétrica, este esquema no necesita que el remitente conozca la clave privada que necesita el destinatario para des-

bloquear el dato. La clave pública se distribuye de modo que cualquiera que quiera iniciar una comunicación segura de datos pueda utilizarla. La clave privada nunca se distribuye; siempre se mantiene en secreto.

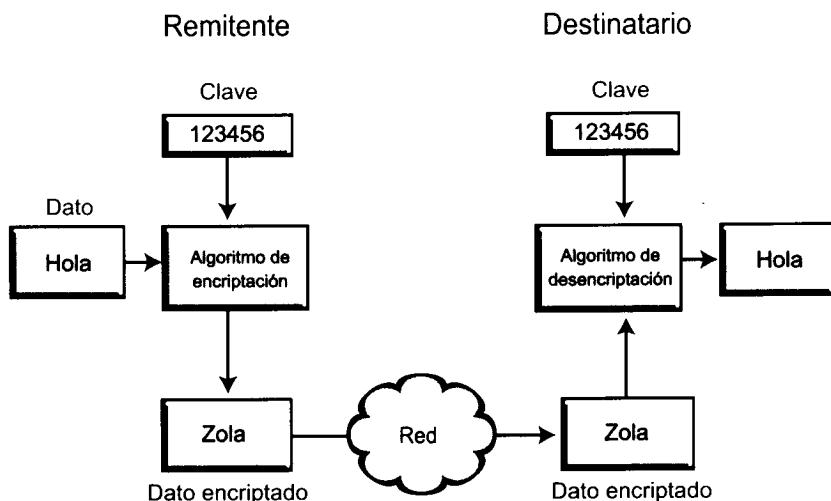


Figura 19.3. Ejemplo de un esquema de encriptación simétrica

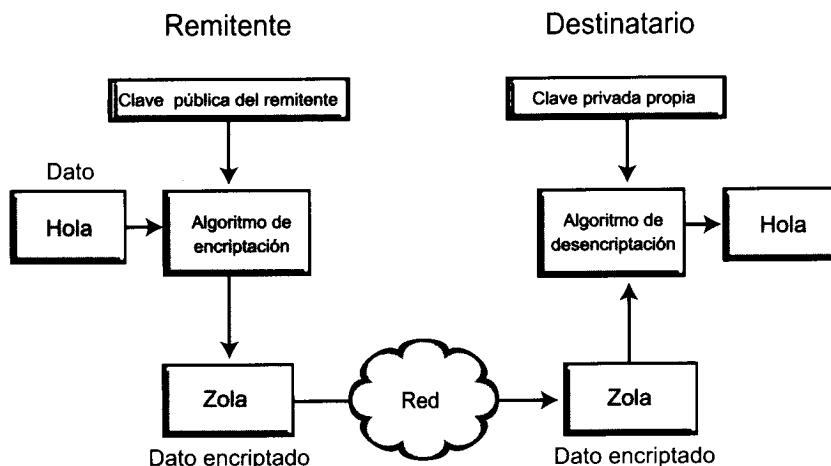


Figura 19.4. Ejemplo de un esquema de una encriptación asimétrica

## Entender los certificados

Un *certificado* encripta información que asocia una clave pública con la verdadera identidad de un individuo, un servidor o de cualquier otra entidad, conocida como el titular. Además incluye la identificación y la firma del emisor del

certificado. El emisor del certificado se conoce con el nombre de Autoridad certificadora (CA). El certificado puede contener otra información, como es el número de serie, el período de tiempo de validez del certificado, y similares; lo que ayuda a la CA a gestionar certificados. Utilizando un navegador Web compatible con SSL, como Netscape Navigator y Microsoft Internet Explorer, puede ver un certificado de servidor fácilmente. La entidad que se identifica en un certificado se representa utilizando campos DN (distinguished name), que se definen en el estándar X509. La tabla 19.1 muestra estos campos.

**Tabla 19.1.** Campos DN

Campo DN:	Abreviatura	Significado
Common Name	CN	Nombre de la entidad que certifica.
Organization/ Company	O	La entidad está asociada con esta organización.
Organizational Unit	OU	La entidad está asociada con esta unidad organizativa.
City/ Locality	L	La entidad está en esta ciudad.
State/Province	ST	La entidad está en este estado o provincia.
Country	C	El nombre se localiza en este país (2-dígitos ISO para el código del país).

El certificado se transmite normalmente en formato binario o de texto codificado.

## Transacciones basadas en certificados

En una transacción basada en SSL, como se muestra en la figura 19.5, el servidor envía un certificado al sistema del cliente.

Las autoridades certificadoras (CA) son las que normalmente emiten los certificados. Estos se encriptan utilizando la clave privada de la CA. El cliente desencripta el certificado utilizando la clave pública de la CA.

Como el certificado contiene la clave pública del servidor, el cliente puede desencriptar y encriptar el dato enviado por el servidor. Entonces, el servidor envía un fragmento de datos identificándose a sí mismo como la identidad mencionada en el certificado. El servidor puede crear una digestión del mensaje del mismo dato que ha enviado para identificarse. La digestión se encripta utilizando la clave privada del servidor. El cliente tiene ahora el certificado de una CA que indica cuál debería ser la clave pública del servidor, un mensaje de identificación

desde el servidor y un mensaje digerido de encriptación del mensaje de identificación.

Utilizando la clave pública del servidor, el cliente puede desencriptar el mensaje digerido. Crea entonces una digestión del mensaje de identificación y lo compara con la digestión enviada por el servidor. Si coinciden, significa que el servidor es quien dice ser. El servidor envía inicialmente un certificado firmado por una CA, de modo que el cliente está totalmente seguro de a quién pertenece la clave pública. Sin embargo, el cliente necesita pruebas de que el servidor que envía el certificado es quien dice ser, de modo que el servidor envía un sencillo mensaje de identificación junto con una digestión encriptada con la clave pública del mismo mensaje. Si el servidor no tiene la clave privada apropiada, será incapaz de producir la misma digestión que el cliente ha realizado para el mensaje de identificación.

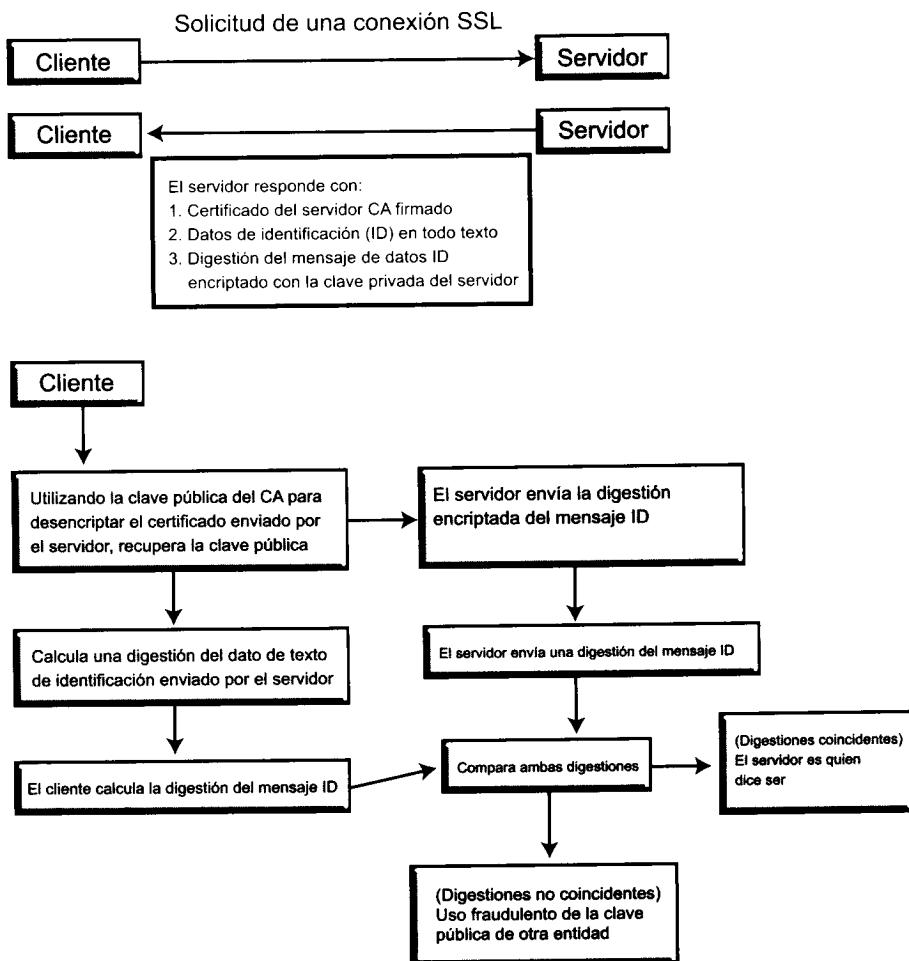


Figura 19.5. Ejemplo de una transacción SSL

Da la sensación de ser un problema complicado y así es, además, la cosa no acaba aquí. El cliente puede enviar una clave de encriptación simétrica al servidor encriptándola con la ayuda de la clave pública del servidor. El servidor puede utilizar esta nueva clave para encriptar datos y transmitírselos al cliente. Pero, ¿para qué hacerlo? Porque la encriptación simétrica es mucho más rápida que la asimétrica. Puede verlo de este modo: la encriptación asimétrica (clave privada/clave pública) se utiliza para transmitir, desde el cliente al servidor con seguridad, una clave generada simétricamente; esta clave se utiliza después para proporcionar un canal de comunicación veloz y seguro.

Si un impostor se sitúa entre el sistema del cliente y del servidor, y es capaz de interceptar los datos que se están transmitiendo, posiblemente sea incapaz de hacer daño. No conoce la clave simétrica secreta que el cliente y el servidor están utilizando, por eso, no puede determinar el contenido del dato; sin embargo, puede introducir basura en el dato inyectando sus propios datos en el paquete de datos.

Para evitar esto, el protocolo SSL permite el código de autentificación de mensajes (MAC) para el usuario. Un MAC es simplemente una pieza de datos que está calculada utilizando una clave simétrica y los datos que se van a transmitir. Como el impostor no conoce la clave simétrica, no puede calcular el valor correcto de MAC. Por ejemplo, se puede utilizar un algoritmo de digestión criptográfico muy conocido llamado MD5, que fue desarrollado por RSA Data Security, Inc., para generar valores MAC de 128-bit para cada paquete de datos que se va a transmitir. Es imposible calcular el poder y el tiempo necesarios para adivinar el valor correcto de MAC. SSL garantiza el comercio seguro en Internet.

## Definir una Autoridad certificadora

Una CA es una organización de confianza que emite certificados para servidores y clientes (es decir, usuarios). Para entender la necesidad de este tipo de organizaciones, considere el escenario siguiente.

Un cliente quiere acceder de forma segura a una aplicación Web del servidor Web de su extranet. El cliente utiliza el protocolo HTTPS para acceder al servidor de su extranet, por ejemplo, <https://extranet.domain.com/login.servlet>.

El navegador Web del cliente inicia la solicitud de la conexión SSL. El servidor Web de su extranet utiliza su clave privada para encriptar los datos que envía al navegador Web del cliente, que desencripta los datos utilizando la clave pública de su servidor Web. Como el servidor Web también envía la clave pública al navegador Web, no hay forma de determinar si la clave pública es auténtica. En otras palabras, no hay nada que pueda evitar que un pirata informático intercepte la información desde el servidor de su extranet y que envíe la clave pública del pirata a su cliente. Aquí es donde entran en juego las CA. Una CA le enviaría un certificado después de comprobar la información sobre su compañía. Este certificado del servidor está firmado por la propia clave pública de la CA, que es

conocida. En este caso, cuando el navegador Web recibe el certificado del servidor, puede desencriptar la información del certificado utilizando la clave pública de la CA. Esto garantiza que el certificado del servidor es auténtico. El navegador Web puede comprobar entonces que el nombre del dominio que se utiliza en el certificado auténtico es el mismo que el nombre del dominio con el que se quiere conectar.

Del mismo modo, si necesita garantizar que el cliente es realmente quien dice ser, podría imponer una restricción de certificado del lado del cliente, lo que le permitiría realizar toda la transacción dentro de bucle cerrado de seguridad.

La idea es que si cada parte tiene un certificado que valida la identidad del otro, confirma la clave pública, y está firmada por una agencia de confianza, ambos saben que se están comunicando con las personas con las que creen que se están comunicando. Hay dos tipos de Autoridades de certificación: CA comerciales y CA privadas de auto certificación.

### **CA comerciales**

El principal trabajo de una CA es verificar a otras compañías comerciales o sin ánimo de lucro que quieren establecer su autenticidad en Internet. Una vez que una CA comprueba la autenticidad de la compañía comprobando determinados registros legales, como los documentos oficiales de registro de la compañía o los contratos, letras oficiales del director de la compañía, y similares, la CA puede firmar el certificado. Hay sólo unas cuantas CA comerciales. Verisign ([www.verisign.com](http://www.verisign.com)) y Thawte ([www.thawte.com](http://www.thawte.com)) son las más conocidas. Verisign ha adquirido Thawte Consulting.

### **CA privadas de auto certificación**

Una CA privada es como la raíz de una CA comercial. Está auto certificada. Sin embargo, una CA privada se utiliza normalmente en un entorno LAN o WAN, o en experimentaciones con SSL. Por ejemplo, una universidad con una WAN que interconecta departamentos podría utilizar una CA privada en lugar de una comercial. Siempre que no espere que usuarios desconocidos confíen en su CA privada, puede utilizarla. En la sección siguiente, le mostraré cómo crear una CA privada para su organización. Después, le enseñaré a obtener un certificado de una CA comercial o a crear su propia CA para certificar sus servidores y clientes.

## **Establecer SSL para Apache**

Tiene dos opciones de código abierto cuando decide utilizar SSL con Apache. Puede utilizar el módulo `mod_ssl` o la distribución Apache-SSL. Cualquiera de estas soluciones necesita OpenSSL, que es una implementación de código abierto de la librería SSL. En esta sección discutiré cómo puede establecer OpenSSL y las soluciones `mod_ssl` y Apache-SSL para Apache.

# Opciones SSL

El proyecto OpenSSL es una colaboración de una comunidad de código abierto que desarrolla SSL, Transport Layer Security (TLS), y paquetes de una librería criptográfica de propósito general. La implementación actual de SSL también se llama OpenSSL. OpenSSL está basado en la librería SSLeay, desarrollada por Eric A. Young y Tim J. Hudson. La licencia del paquete de software OpenSSL permite que se utilice, de forma gratuita, el software tanto para propósitos comerciales como no comerciales.

Hay dos paquetes OpenSSL disponibles de forma gratuita para utilizar con Apache:

- Los parches Apache-SSL basados en OpenSSL, que puede encontrar en [www.apache-ssl.org](http://www.apache-ssl.org).
- El módulo `mod_ssl` basado en OpenSSL para Apache, que puede encontrar en [www.modssl.org](http://www.modssl.org).

Cualquiera de las dos opciones aportan la misma funcionalidad. Algunas personas prefieren Apache-SSL y otras prefieren `mod_ssl`. Ambas funcionan igual pero difieren en los estilos de código, documentación y en otras características menores. Yo me decanto por `mod_ssl` porque me gustan otros módulos de Ralf S. Engelhall, como el módulo `mod_rewrite`.

## Establecer OpenSSL

El paquete OpenSSL, que encontramos en [www.openssl.org](http://www.openssl.org), es necesario para que las soluciones Apache-SSL y `mod_ssl` sean capaces de aportar SSL al servidor Web de Apache. En esta sección, aprenderá a establecer OpenSSL en su sistema Unix.

Voy a utilizar un sistema Linux para aclararlo. El sitio Web OpenSSL ofrece el código OpenSSL en un archivo tar comprimido en gzip. La última versión es `openssl-0.9.6.tar.gz`.

### Requisitos previos de OpenSSL

Antes de comenzar el proceso de compilación, debe asegurarse de que su sistema cumple los requisitos previos de OpenSSL. La distribución fuente de OpenSSL exige que tenga Perl 5 y un compilador de ANSI C. Este capítulo supone que tiene instalados Perl 5 y gcc (compilador de C) cuando establece su sistema Linux.

### Obtener OpenSSL

SSL ha estado disponible en el software comercial Linux como Stronghold, y el servidor Web comercial basado en Apache durante muchos años. Sin embargo, debido a algunas patentes y a restricciones de exportación en U.S., no ha habido

versiones de SSL de código abierto para Linux durante mucho tiempo. El proyecto Open SSL proporciona una versión para Linux de SSL, que puede bajar del sitio Web oficial de OpenSSL en [www.openssl.org/source](http://www.openssl.org/source).

**TRUCO:** La distribución de Red Hat Linux contiene binarios OpenSSL en paquetes RPM. De modo que también puede utilizar la versión RPM suministrada por Red Hat Linux o puede bajar la distribución de la fuente, compilarla e instalarla.

Yo prefiero que el software de seguridad se instale desde la distribución de la fuente cargada desde los sitios Web o FTP auténticos. Supongo que prefiere hacer lo mismo. Por eso, la siguiente sección discute los detalles de la compilación e instalación de OpenSSL desde la distribución fuente bajada del sitio Web de OpenSSL.

**NOTA:** Si tiene que instalar OpenSSL desde el RPM, utilice una distribución RPM binaria de confianza como la que se encuentra en el CD-ROM oficial de Red Hat. Para instalar los binarios OpenSSL desde un paquete RPM, ejecute el comando `rpm -ivh openssl-packagename.rpm`.

## Compilar e instalar OpenSSL

Compilar OpenSSL es una tarea muy sencilla, simplemente siga los pasos siguientes:

1. Regístrese en su sistema Linux como raíz desde la consola.
2. Copie la fuente "tar ball" de OpenSSL en el directorio `/usr/src/redhat/SOURCES`.
3. Extraiga la distribución fuente ejecutando el comando `tar xvzf openssl-version.tar.gz`. Por ejemplo, para extraer el archivo `openssl-0.9.6.tar.gz`, ejecute el comando `tar xvzf openssl-0.9.6.tar.gz`. El comando `tar` crea un directorio llamado `openssl-version`, el cual, en mi ejemplo, es `openssl-0.9.6`.
4. Puede eliminar el "tar ball" en este momento si el espacio en el disco es un problema. Sin embargo, le recomiendo que lo elimine únicamente después de compilar e instalar OpenSSL.
5. Cambie su directorio actual para que sea el nuevo directorio creado.

Ahora puede leer los archivos `README` o `INSTALL` incluidos en la distribución. Antes de compilar el software, debería configurar las opciones de configura-

ción. Para instalar OpenSSL en el directorio por defecto /usr/local/ssl, ejecute:

```
./config
```

Sin embargo, si tiene que instalarlo en un directorio distinto, adjunte los indicadores --prefix y --openssldir al comando anterior. Por ejemplo, para instalar OpenSSL en el directorio /opt/security/ssl, la línea de comandos es:

```
./config --prefix=/opt/security
```

Hay muchas otras opciones que puede utilizar con el script config o el Configure, para preparar la distribución fuente para compilación. Estas opciones se discuten en la tabla 19.2.

**Tabla 19.2.** Opciones de configuración para compilar OpenSSL

Opciones de configuración	Propósito
--prefix=DIR	Instala OpenSSL en el directorio DIR. Se crean subdirectorios como DIR/lib, DIR/bin, DIR/include/openssl. Los archivos de configuración se almacenan en DIR/ssl a no ser que utilice la opción --openssldir para especificar este directorio.
--openssldir=DIR	Especifica el directorio de los archivos de configuración. Si no se utiliza la opción --prefix, todos los archivos se almacenan en este directorio.
Rsaref	Esta opción forzará a la construcción del kit de herramientas RSAREF. Si quiere utilizar el kit de herramientas RSAREF, asegúrese de tener la librería RSAREF (librsaref.a) en su ruta de búsqueda de librerías por defecto.
no-threads	Esta opción desactiva el soporte para aplicaciones multihilo.
threads	Esta opción activa el soporte de aplicaciones multihilo.
no-shared	Esta opción desactiva la creación de una librería compartida.
Shared	Esta opción activa la creación de una librería compartida.

Opciones de configuración	Propósito
no-asm	Esta opción desactiva la utilización de código ensamblador en el árbol fuente. Utilice esta opción únicamente si ha tenido problemas en la compilación de OpenSSL.
386	Utilice esta opción únicamente si está compilando OpenSSL en una máquina Intel 386. No está recomendado para las máquinas Intel nuevas.
no-<cipher>	OpenSSL utiliza muchos cifrados criptográficos como bf, cast, des, dh, dsa, hmac, md2, md5, mdc2, rc2, rc4, rc5, rsa y sha. Si no quiere incluir un cifrado determinado en los binarios compilados, utilice esta opción.
-Dxxx, -Ixxx, -Lxxx, -fxxx, -Kxxx	Estas opciones le permiten especificar varias opciones dependientes del sistema, por ejemplo, se pueden especificar indicadores Dynamic Shared Objects (DSO) en la línea de comando, como -fpic, -fPIC y -KPIC. De este modo, se pueden compilar las librerías OpenSSL con Position Independent Code (PIC), que son necesarias para enlazarlas en DSO.

Lo más probable es que no necesite añadir ninguna de estas opciones para compilar OpenSSL. Sin embargo, si tiene problemas compilando, debería intentar alguna de estas opciones con los valores apropiados. Por ejemplo, si no lo puede compilar porque OpenSSL no encuentra los archivos library, intente especificar la ruta de las librerías de sistema utilizando la opción -L.

Una vez que ha ejecutado el script config sin ningún error, ejecute la utilidad make. Si el comando make tiene éxito, ejecute make test para comprobar si hay binarios nuevos. Finalmente, ejecute make install para instalar OpenSSL en su sistema. Si tiene algún problema durante la compilación, intente comprender la causa del problema. En la mayoría de los casos, el problema está causado porque el archivo library no coincide cuando tratamos de instalar la última versión del software OpenSSL en un sistema Linux antiguo. El problema también puede ser causado por alguna opción que especificó en la línea de comandos. Por ejemplo, si no tiene la librería RSAREF (que no está incluida en la distribución de Red Hat Linux) instalada en su sistema y está tratando de utilizar la opción rsaref, la compilación fallará cuando intente construir los binarios.

Por lo tanto, tiene que saber lo que está haciendo cuando utiliza determinadas opciones. Si sigue sin resolver el problema, intente realizar una búsqueda en la página FAQ de OpenSSL en [www.openssl.org/support/faq.html](http://www.openssl.org/support/faq.html) para encontrar una solución. O, simplemente instale el paquete RPM binario para OpenSSL.

## Elegir el módulo mod\_ssl para soporte SSL

Debería utilizar mod\_ssl únicamente si decide no utilizar Apache-SSL para aportar soporte SSL. Este módulo se puede utilizar como un módulo DSO para Apache y además funciona con la plataforma Windows.

### Compilar e instalar mod\_ssl

Para compilar e instalar este módulo, siga los siguientes pasos:

1. Baje la última distribución fuente del módulo mod\_ssl de [www.modssl.org](http://www.modssl.org). Extraiga la fuente en un directorio.
2. Como raíz, ejecute el script `Configure` desde el nuevo directorio `mod_ssl-version` creado.

```
./configure --with-apache=../httpd_version
```

Asegúrese de reemplazar `../httpd_version` con la ruta apropiada para el directorio de la distribución fuente de Apache. Por ejemplo, si instaló la fuente mod\_ssl en el directorio `/usr/local/src/mod_ssl-2.9.0-2.0.19`, y la distribución fuente de Apache en el directorio `/usr/local/src/httpd_2.0.19`, entonces ejecute el siguiente comando desde el directorio `/usr/local/src/mod_ssl-2.9.0-2.0.19`:

```
./configure --with-apache=../httpd_2.0.19
```

3. Asigne una variable de entorno llamada `SSL_BASE` al directorio OpenSSL en su sistema. Por ejemplo, si está utilizando el shell bash puede ejecutar `SSL_BASE=path_to.openssl`, donde `path_to.openssl` debería reemplazarse con la ruta actual para el directorio de instalación de OpenSSL.
4. Diríjase al directorio fuente de máximo nivel de Apache y ejecute el script `configure` utilizando la opción `--enable-module=ssl` junto con el resto de las opciones habituales necesarias para Apache. Si ya tiene compilado Apache, puede utilizar `config.status --enable-module=ssl` para reutilizar todas las opciones previas de la línea de comando de forma automática.

**TRUCO:** Si ha pensado utilizar mod\_ssl como módulo compartido, entonces utilice --with-apxs --enable-shared=ssl -enable-so en lugar de --enable-module=ssl con el script configure.

5. Compile e instale el servidor Apache con soporte mod\_ssl utilizando el comando make.
6. Si no tiene un certificado del servidor, puede crear un certificado de ensayo utilizando el comando make certificate TYPE=dummy.
7. Ejecute make install para instalar el servidor Apache compatible con mod\_ssl. Si ha creado el certificado del servidor en el paso 6, se copiará en el subdirectorio conf de la ruta de la instalación de su servidor Apache.
8. Si se está estableciendo Apache, párelo utilizando el comando /usr/local/bin/apachectl stop y reinicie Apache con soporte para HTTPS utilizando el comando /usr/local/bin/apachectl sslstart startssl.

## Configurar Apache para SSL basado en mod\_ssl

Si compila mod\_ssl con Apache una vez que ya tiene construido Apache, las directivas de configuración relacionadas con SSL estarán almacenadas en httpd.conf.default en lugar de en httpd.conf. Básicamente, cuando añade mod\_ssl a un servidor Apache existente, se conserva el archivo httpd.conf. Debe configurar de forma manual las directivas relacionadas con SSL copiando la información necesaria de httpd.conf.default. En el material siguiente, he supuesto que ha añadido mod\_ssl al sistema Apache volviendo a compilar Apache con soporte mod\_ssl, tal y como se mostró en la sección anterior. Para configurar Apache para SSL utilizando mod\_ssl, siga los pasos siguientes:

1. Si instala mod\_ssl como un módulo, entonces añada la siguiente línea en httpd.conf:  
`AddModule ssl_module modules/ssl/libssl.soa`
2. Modifique httpd.conf de modo que incluya las directivas siguientes:

```
<IfDefine SSL>
    Listen 80
    Listen 443
</IfDefine>
```

Este contenedor <IfDefine> le dice a Apache que considere las directivas encerradas sólo si Apache se inicio con la opción -DSSL o si se utilizó apachectl sslstartstartssl. En el modo SSL, se le pide a

Apache que escuche las conexiones en el puerto estándar HTTP, el puerto 80, y también en el puerto HTTPS estándar 443.

3. Añada las líneas siguientes en httpd.conf para decirle a Apache que añada dos tipos MIME para las extensiones.crt y .crl. La primera extensión indica un archivo de certificación y el segundo indica un archivo con una lista de revocación de certificados (CRL). Estas dos líneas garantizan que Apache envía la cabecera cuando envía archivos con estas extensiones.

```
<IfDefine SSL>
    AddType application/x-x509-ca-cert .crt
    AddType application/x-pkcs7-crl     .crl
</IfDefine>
```

4. Añada las líneas siguientes a httpd.conf para definir la ruta de la base de datos en el caché de sesión SSL, el valor del tiempo de expiración del temporizador, la ruta del archivo SSL mutex (o controlador de acceso a recursos) utilizado por los hijos activadores de SSL para comunicarse, el método de generación al azar SSL, la ruta de registro SSL, y la información del nivel de registro SSL.

```
<IfModule mod_ssl.c>

    SSLPassPhraseDialog builtin

    SSLSessionCache dbm:/usr/local/apache/logs/ssl_scache
    SSLSessionCacheTimeout 300

    SSLMutex file:/usr/local/apache/logs/ssl_mutex

    SSLRandomSeed startup builtin
    SSLRandomSeed connect builtin

    SSLLog /usr/local/apache/logs/ssl_engine_log
    SSLLogLevel info

</IfModule>
```

5. Tiene que crear una configuración por defecto del host virtual en httpd.conf, tal y como se observa a continuación:

```
<IfDefine SSL>

    <VirtualHost _default_:443>

        DocumentRoot path_to_ssl_document_root
        ServerName ssl_server_hostname

        SSLEngine on
```

```

SSLCipherSuite
ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:
+SSLv2:+EXP:+eNULL
    SSLCertificateFile conf/ssl.crt/server.crt

    SSLCertificateKeyFile conf/ssl.key/server.key

</VirtualHost>

</IfDefine>

```

Este host virtual `_default_` impide que el servidor principal sirva ninguna solicitud SSL (en el puerto 443). Atrapa todas las solicitudes SSL que no se pueden integrar con ningún otro host virtual definido en `httpd.conf`.

- Debería cambiar `path_to_ssl_document_root` a la raíz de documentos apropiada para el servidor SSL. Puede utilizar la misma raíz de documentos que utilizó para su sitio HTTP (puerto 80).
  - El nombre del servidor `ssl_server_hostname` tiene que ser el apropiado. Puede utilizar el mismo nombre de host para HTTP y para HTTPS.
  - La directiva `SSLEngine` activa el módulo SSL para este host virtual.
  - `SSLCipherSuite` define el cifrado criptográfico que se puede utilizar en este host virtual.
  - La directiva `SSLCertificateFile` define la ruta del archivo de certificado del servidor. Asegúrese de que la ruta es correcta para su sitio.
  - `SSLCertificateKeyFile` define el archivo clave para el certificado. De nuevo, asegúrese de que ha definido la ruta adecuada.
6. Si ejecuta scripts `mod_perl`, scripts CGI, SSI, o páginas PHP, y quiere tener acceso a las variables de entorno estándar relacionadas con SSL/TLS (prefijadas por `SSL_`), entonces debe instruir explícitamente a Apache para que cree estas variables para solicitudes `mod_perl`, CGI, SSI, o PHP. Utilice `SSLOptions +StdEnvVars` para decirle a Apache que genere las variables de entorno `SSL_*` para estas solicitudes. Sin embargo, es una buena idea activar esta opción en el contenedor `<File>`, tal y como se muestra a continuación:

```

<Files ~ "\.(pl| cgi|shtml|php)$">
    SSLOptions +StdEnvVars
</Files>

```

Las variables de entorno `SSL_*` se han creado únicamente para archivos con extensiones `.pl` (scripts Perl), `.cgi` (scripts CGI), `.shtml` (páginas SSI) o `.php` (páginas PHP). Puede situar este contenedor `<File>` dentro del host virtual definido en el paso 5.

7. Del mismo modo, si utiliza el alias estándar /cgi-bin/ para los scripts CGI, puede activar la creación del entorno SS\_\* utilizando el contenedor <Directory>:

```
<Directory "physical_path_to_cgi_bin_directory">
    SSLOptions +StdEnvVars
</Directory>
```

Asegúrese de reemplazar physical\_path\_to\_cgi\_bin\_directory con la ruta real del directorio CGI.

8. Inicie Apache utilizando el comando /usr/local/apache/bin/apachectl sslstartstartssl.

## Elegir Apache-SSL en lugar de mod\_ssl para soporte SSL

El kit Apache-SSL se puede bajar del sitio Web [www.apache-ssl.org](http://www.apache-ssl.org). El kit de la fuente Apache-SSL convierte Apache en un servidor SSL basado en SSLeay o en OpenSSL.

### Compilar e instalar parches Apache-SSL para Apache

Tiene que asegurarse de que ha instalado OpenSSL en su sistema. El siguiente material discute cómo compilar e instalar Apache con parches Apache-SSL.

Voy a suponer que tiene instalado OpenSSL en el directorio /usr/local/ssl y que quiere extraer el árbol fuente de Apache en el directorio /usr/local/src/httpd\_version. Por ejemplo, la ruta de la fuente Apache para Apache 2.0.16 debería ser /usr/local/src/httpd\_2.0.16. Para establecer Apache con soporte SSL, siga los siguientes pasos:

1. Como usuario raíz cambie el directorio al directorio de la distribución fuente de Apache (/usr/local/src/httpd\_version).
2. Copie el archivo con el kit de parches Apache-SSL (apache\_version+ssl\_version.tar.gz) en el directorio actual y extráigalo utilizando el comando tar xvzf apache\_version+ssl\_version.tar.gz.
3. Ejecute patch -p1 < SSLpatch para parchear los archivos fuente.
4. Cambie el directorio a src y edite el archivo Configuration.tmpl para tener las líneas siguientes junto con el resto de líneas conservadas.

```
SSL_BASE=/usr/local/ssl
SSL_APP_DIR= $(SSL_BASE)/bin
SSL_APP=/usr/local/ssl/bin/openssl
```

5. Vuelva a cambiar el directorio actual a src ejecutando el comando cd...

6. Ejecute el comando `./configure` con cualquier argumento de la línea de comandos que utilice normalmente. Por ejemplo, si quiere instalar Apache en `/usr/local/apache`, ejecute este script con la opción `--prefix=/usr/local/apache`.
7. Ejecute el comando `make && make install` para compilar e instalar Apache.

Estos pasos compilan e instalan Apache estándar (`httpd`) y Apache para SSL (`httpsd`). Ahora tiene que crear un certificado para el servidor Apache.

## Crear un certificado para el servidor Apache-SSL

Para crear un certificado temporal, siga estos pasos:

1. Cambie el directorio al subdirectorio `src` (por ejemplo `/usr/local/src/httpd_version/src`) de su distribución fuente de Apache.
2. Una vez que se encuentra en el directorio `src`, ejecute el comando `make certificate` para crear un certificado temporal de prueba. El comando `make certificate` utiliza el programa `/usr/local/ssl/bin/openssl` para crear un certificado del servidor. Le hará una serie de preguntas. A continuación tiene un ejemplo de este comando:

```
ps > /tmp/ssl-rand; date >> /tmp/ssl-rand; \
RANDFILE=/tmp/ssl-rand /usr/local/ssl/bin/openssl req - \
config ../SSLconf/conf/ssleay.cnf \
-new -x509 -nodes -out ../SSLconf/conf/httpsd.pem \
-keyout ../SSLconf/conf/httpsd.pem; \
ln -sf httpsd.pem ../SSLconf/conf/'/usr/local/ssl/bin/ \
openssl \
x509 -noout -hash < ../SSLconf/conf/httpsd.pem'.0; \
rm /tmp/ssl-rand
Using configuration from ../SSLconf/conf/ssleay.cnf
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to '../SSLconf/conf/httpsd.pem'
----
```

Le pedirá que introduzca información en su solicitud de certificado.

Va a introducir los llamados Distinguished Name o DN. Hay muchos campos pero puede dejar algunos en blanco. Algunos campos tienen un valor por defecto, Si introduce '.', el campo se quedará en blanco.

```
-----
Country Name (2 letter code) [GB]:US
State or Province Name (full name) [Some-State]:California
Locality Name (eg, city) []:Sacramento
Organization Name (eg, company; recommended) []:MyORG
```

```
Organizational Unit Name (eg, section) []:CS
server name (eg. ssl.domain.tld; required!!!!)
[]:shea.evoknow.com
Email Address []:kabir@evoknow.com
```

Se crea el certificado httpsd.pem en el subdirectorio SSLconf/conf de la distribución fuente de Apache.

Ahora está preparado para configurar Apache.

## Configurar Apache con Apache-SSL

Cuando ejecutó make install en la sección "Compilar e instalar parches Apache-SSL para Apache", se creó un archivo httpsd.conf en el subdirectorio conf del directorio de instalación de Apache. Por ejemplo, si utiliza --prefix=/usr/local/apache para configurar Apache, encontrará el archivo httpsd.conf en el directorio /usr/local/apache/conf. Tiene que volver a nombrarlo para httpd.conf, utilizando el comando mv /usr/local/apache/conf/httpd.conf /usr/local/apache/conf/httpd.conf. Reemplace /usr/local/apache/conf con el nombre de ruta apropiado, en el caso de que instalase Apache en un directorio distinto.

Tiene dos opciones cuando va a utilizar SSL con Apache. Puede activar SSL para el servidor principal de Apache, o puede activarlo para uno o más sitios Web virtuales. A continuación veremos cómo activar SSL para el servidor principal de Apache y también discutiremos lo que necesita para activar SSL para un sitio Web virtual. Siga los siguientes pasos para modificar el archivo httpd.conf:

1. Por defecto, los navegadores Web enviarán solicitudes SSL al puerto 443 de su servidor Web, por lo tanto, si quiere convertir el servidor Apache principal en un servidor para SSL, cambie la línea de la directiva Port a:  
Port 443
2. Añada las líneas siguientes para decirle a Apache cómo generar datos al azar necesarios para las conexiones SSL encriptadas:

```
SSLRandomFile file /dev/urandom 1024
SSLRandomFilePerConnection file /dev/urandom 1024
```

3. Si quiere rechazar todas las solicitudes excepto las solicitudes https (es decir, las solicitudes SSL), inserte la directiva siguiente:

```
SSLRequireSSL
```

4. Para activar el servicio SSL, añada la siguiente directiva:

```
SSLEnable
```

5. Por defecto, el servidor caché utilizado por Apache con SSL, se crea en el directorio src/modules/ssl de la distribución fuente de Apache. Determine este directorio del siguiente modo:

```
SSLCacheServerPath \
/path/to/apache_version/src/modules/ssl/gcache
```

6. Añada las siguientes directivas para activar los valores del puerto del servidor caché y del tiempo de expiración del temporizador del caché:

```
SSLCacheServerPort logs/gcache_port
SSLSessionCacheTimeout 15
```

7. Ahora tiene que decirle a Apache dónde está guardando el archivo del certificado. Si creó el certificado del servidor siguiendo las instrucciones de instalación de OpenSSL, el certificado del servidor debería estar en el directorio /usr/local/ssl/certs. Sin embargo, si decidió utilizar temporalmente el certificado de prueba (que se creó utilizando el comando make certificate), entonces su certificado de prueba estará en el directorio /path/to/apache\_version/SSLconf/conf y se llamará httpsd.pem. Asigne la directiva siguiente a la ruta completa del certificado de su servidor, tal y como se muestra a continuación:

```
SSLCertificateFile \
/path/to/apache_version/SSLconf/conf/httpsd.pem
```

8. Para terminar, asigne las directivas siguientes y guarde el archivo httpd.conf:

```
SSLVerifyClient 3
SSLVerifyDepth 10
SSLFakeBasicAuth
SSLBanCipher NULL-MD5:NULL-SHA
```

9. Para activar SSL en un host virtual llamado myvhost.Evoknow.com en el puerto 443, necesito la siguiente configuración:

```
Listen 443
<Virtualhost myvhost.evoknow.com:443>
  SSLEnable
  SSLCertificateFile /path/to/myvhost.certificate.cert
</Virtualhost>
```

Su servidor compatible con SSL está listo para que lo pruebe.

## Probar su conexión SSL

Una vez que inicia el servidor Apache con el comando /usr/local/apache/bin/apachectl sslstartstartssl, debería ser capaz de acceder al sitio SSL utilizando https://localhost/ o https://your\_server\_hostname/ desde clientes locales (en el propio servidor Web) o remotos.



(normalmente desplegado por un pequeño icono que representa una cerradura en la barra de estado de los navegadores Web).

## Obtener un certificado

Antes de que pueda utilizar Apache con SSL (mediante mod\_ssl o mediante Apache-SSL), debe crear el certificado apropiado para el servidor. Puede obtener un certificado para el servidor de una autoridad de certificación comercial o puede crear su propia autoridad de certificación y entonces crear certificados para sus propios servidores y para sus clientes. El último método se utiliza normalmente en grandes organizaciones que quieren gestionar certificados entre sus distintos campos, de forma interna.

### Obtener un certificado para el servidor desde una CA comercial

Para obtener un certificado firmado por una CA comercial, debe cumplir una serie de requisitos. Hay dos tipos de requisitos:

Usted (el solicitante) debe probar que es la entidad que dice ser.

Debe presentar una solicitud de certificación firmada, Certificate Signing Request (CSR), en formato electrónico.

El primer requisito se cumple siguiendo las guías de la CA para la comprobación de individuos y organizaciones, de modo que debe consultar con la CA elegida para conocer el procedimiento. Normalmente, si ha decidido obtener un certificado para su servidor Web, ha de estar preparado para presentar copias de documentos legales, como el registro de negocio o contratos. También puede preparar un CRS utilizando OpenSSL.

El primer paso para la creación de un CSR es crear una clave privada para su servidor. Tendrá que crear una solicitud firmada de un certificado que tiene que enviar a la autoridad de certificación comercial. Una vez que la autoridad de certificación ha aprobado su solicitud, podrá instalar el certificado en su servidor y utilizar SSL con Apache. Estos detalles se discuten en las secciones siguientes.

### Generar una clave privada

Para generar una clave privada encriptada para un servidor Web llamado www.domain.com ejecute:

```
openssl genrsa -des3 -out www.domain.com.key 1024 -rand /dev/urandom
```

Cuando ejecute este comando, le pedirá que introduzca una frase de paso (es decir, una contraseña) para encriptar la clave privada. Como la clave privada se encripta utilizando el cifrador des3, le pedirá que introduzca una frase de paso cada vez que se inicie su servidor. Si esto no le gusta, puede crear una versión sin encriptar de la clave privada eliminando la opción -des3 de la línea de comandos. Le recomiendo que utilice una clave privada encriptada para garantizar un alto nivel de seguridad. después de todo, no querrá que nadie que pueda entrar en su servidor sea capaz de ver su clave privada. El listado 19.1 muestra el contenido del archivo www.domain.com.key.

#### Listado 19.1. El contenido del archivo www.domain.com.key

```
----BEGIN RSA PRIVATE KEY----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,C48E9F2F597AF968

47f4qGkVrfFFtNEygEs/uyaPOeAqksOnALtKUvADHKL7BhaB+8BrT/
Haa7MHwEzU
jjaRd1XF1k1Ej3qh6d/
Z10AwVfYiAYvO1H3wQB2p11Suxui2sm72RKYUOpRMjxZI
/srHn/
DU+dUql1pH3vJRW2hHNvjHUB0cuCsz28GOhICa5MFGsZxDR+cKP0T2Uvf5
jlGyiMroBzN0QF0v8sqwZoSOSuKHU9ZKdA/
Pcbu+fwyDWFzNfr8HPNTImlaMjGET
i9LWZikzBW2mmaw79Pq6xSyqL+7dKXmiQL6d/
bYiH0ZUYhMkJtqUp1fNXxJd4T6
KB8xVbjPivo1AyvYK0qmmVQp7WDnEyrrYUZVYRu0a+1050aTG2GnfSy32YGuNTY
1MB3PH5BuocSRp+9SsKKTVoW0a01n0RtgVk/
EZTO2Eo94qPcsZes6YyAwY4ffFVAw
gG/G3ZJCPdjBI2YLmvhua3bvp9duc5CXmKDxO049VvjbEB/
yvi9pLbuj8KuAt4ht
fZcZB94wxrR/EMGODs2xgNhH+SwEf5Pc/bPUMRCq/0t6F/
HJ47jVnUf17tdtOTT7
UbQQVyAsr9tKSFzsRKMOGB04VoenkD5CzUUf3iO/NaXSs/
EFu9HG1ctWRKZEViP/
MSJBe3jYDXbmeGdQGNJUExpY64hv1XoNd0pAJk0E622o2allraFusl2PotNvWYdi
TShgoIHSmNgQQLCfssJH5TABKyLejsgQy5Rz/
Vp3kDzkWhwEC0hI42p0S8sr4GhM
6YEdASb51uP3ftn2ivKshueZHFOvS1pCGjnEYAEEdY4QLJkreznM8w==

----END RSA PRIVATE KEY----
```

## Generar un CSR

Tiene que generar el CSR utilizando su clave privada:

```
openssl req -new -key www.domain.com.key -out www.domain.com.csr
```

En el caso de que encriptase la clave privada, le pedirá que introduzca la frase de paso para la clave privada. Introduzca la frase de paso apropiada. Entonces, le

pedirá que el nombre del país, el estado, la ciudad, el nombre de la organización, el nombre del departamento o de la unidad de la organización, el nombre común (es decir, su nombre, si la solicitud del certificado es para usted, o el nombre del host de su servidor), la dirección de correo electrónico y alguna información adicional como la contraseña y opcionalmente el nombre de la compañía.

En este momento, tiene que presentar su CSR a una CA como Thawte. Como el proceso de certificación implica verificación de sus documentos de identidad individuales o de negocio, tardará algunos días o semanas, incluso meses, en recibir su certificado. La siguiente sección utiliza Thawte como la CA elegida.

Si tiene que obtener el certificado para empezar a probarlo o tiene alguna otra razón por la que necesita un certificado temporal inmediatamente, pídaselo a su CA. Podrían tener algún modo de proporcionarle uno. Por ejemplo, Thawte le dejará presentar su CSR mediante la Web para obtener un certificado temporal, que recibirá al cabo de unos minutos por correo electrónico.

Una vez que ha obtenido el certificado para el servidor de una CA comercial, puede instalar el archivo del certificado con las instrucciones que le facilitarán. Este paso suele ser muy sencillo. Probablemente le pedirán que copie el archivo en un directorio y que reinicie el servidor.

Si no está interesado en obtener un certificado firmado de una CA comercial, puede crear su propia CA y certificar entidades como servidores o usuarios. En la sección siguiente se muestra cómo hacerlo.

## Crear una autoridad de certificación privada

Tal y como se ha indicado, los certificados privados auto firmados no son adecuados para utilizarlos en Internet, en el sentido de que los usuarios no confiarán a la larga en ellos. Sin embargo, si quiere ser capaz de emitir certificados internos para su compañía y no quiere pasar por la verificación de una CA, debe utilizar una CA privada.

**NOTA:** Podría ser posible obtener un certificado de enlace cruzado para su CA privada desde una CA comercial. En este caso, su CA privada se encadenaría a la CA comercial y de ese modo todo el mundo confiaría en los certificados que emitiese. Sin embargo, la CA comercial podría limitar su autoridad de concesión de certificados a su propia organización para que su organización no constituya una competencia para ella.

Siga los pasos siguientes para crear una CA privada de auto certificación utilizando OpenSSL:

1. Baje la última versión de la distribución del script `ssl.ca-version.tar.gz` desde la sección de software de contribución al usuario ([www.openssl.org/contrib](http://www.openssl.org/contrib)) del sitio Web de OpenSSL. Extraiga

este archivo en el directorio de su elección. Se creará un subdirectorio llamado `ssl.ca-version`. Encontrará un conjunto de scripts `sh` en el directorio.

2. Ejecute el script `new-root-ca.sh` para crear un certificado de ruta para su CA privada. Le pedirá que introduzca la frase de paso. Esta frase de paso se necesita para firmar certificados futuros.
3. Para crear un certificado para el servidor, ejecute el script `new-server-cert.sh www.domain.com` para crear una clave privada y una clave pública para el servidor. Le pedirá que introduzca campos DN para el nuevo certificado del servidor. El script generará también un CSR, que puede enviar si quiere a la CA.
4. Ejecute el script `sign-server-cert.sh` para aprobar y firmar el certificado que ha creado utilizando el script `new-server-cert.sh`.
5. Ejecute el script `new-user-cert.sh` para crear un certificado de usuario. Los certificados de usuarios firmados por una autoridad de certificación comercial, se pueden utilizar con el navegador Web para autenticar usuarios en servicios remotos. Sin embargo, los certificados de usuarios no son habituales debido a la carencia de conocimientos sobre ellos y a la falta de software de cliente y servidor disponible.
6. Ejecute el script `sign-user-cert.sh` para firmar un certificado de usuario. Además, ejecute el script `p12.sh` para empaquetar la clave privada, la clave firmada y la clave pública de la CA en un archivo con extensión `.p12`. Este archivo puede importarse dentro de aplicaciones en correos electrónicos para utilizarlas.

Ahora está preparado para utilizar OpenSSL con distintas aplicaciones. OpenSSL es una parte integral de la seguridad. Cuanto más utilice OpenSSL, más fácil encontrará incorporarlo en distintos servicios.

## Acceder a páginas SSL

Si instaló Apache en el directorio `/usr/local/apache`, ejecute el comando `/usr/local/apache/bin/httpsdctl start` para iniciar el servidor Apache compatible con SSL. Si obtiene un mensaje de error, busque los detalles en el archivo de registro.

Una errata o una ruta desaparecida en el archivo `httpd.conf` es una causa habitual, por lo que tiene que analizar este archivo. Cuando se inicia el servidor, puede acceder a él utilizando el protocolo HTTPS. Por ejemplo, para acceder a un servidor Apache con SSL llamado `shea.evoknow.com`, puedo dirigir un navegador Web a `https://shea.evoknow.com`. Si está utilizando el certifi-

cado de prueba o un certificado firmado por una CA, el navegador Web mostrará un mensaje de advertencia indicando que el certificado no se puede verificar. Esto es normal porque ninguna CA conocida ha firmado el certificado. Debería aceptar el certificado y navegar por el sitio Web compatible con SSL.

# Parte V

# Ejecutar

# Apache

# en Win32



# 20 Instalar y ejecutar Apache para Windows

---

## En este capítulo

1. Nombramos los requisitos del sistema para ejecutar Apache en Windows.
2. Instalamos Apache en Windows.
3. Iniciamos, paramos y reiniciamos Apache.

Aunque Apache se ha estado trasladado a la plataforma Windows durante mucho tiempo, Apache 2.0 es la primera versión que puede sacar partido de las llamadas al sistema Windows nativas. Todas las versiones anteriores de Apache utilizaban una capa de abstracción POSIX que disminuía el rendimiento cuando se ejecutaba Apache en Windows. Ahora Apache puede, por fin, competir con otros servidores nativos de Web como el servidor Microsoft IIS y el servidor Netscape Enterprise.

En este capítulo, conocerá los requisitos del sistema para ejecutar Apache bajo plataformas Windows conocidas, como son Windows XP, Windows 2000, Windows NT y Windows 9x/ME; también aprenderá a instalar Apache y a configurarlo de modo que pueda prepararlo y ejecutarlo rápidamente.

# Requisitos del sistema

Apache exige que tenga activado TCP/IP en su sistema Windows. Debería asegurarse de que tiene la última versión del TCP/IP (Winsock) para su plataforma Windows. Los usuarios de Windows 2000 con TCP/IP activado pueden instalar Apache sin ningún otro requisito específico para Windows. Los usuarios de Windows NT 4.0 tienen que tener instalados los paquetes con los últimos servicios. El servidor Windows 2000 Advanced Server es la plataforma ideal para ejecutar un servidor Web de Apache en grado de producción. Windows 2000 Advanced Server está ajustado para servicios de red, de modo que debería producir el mejor rendimiento para Apache junto con otros servicios de Internet.

**NOTA:** Utilice el Windows 2000 Advanced Server como plataforma Windows para los capítulos específicos de Windows de este libro.

También necesita la versión 1.10 del Microsoft Installer, o una versión superior para instalar Apache. Windows 2000 y Windows ME tienen integrado el Microsoft Installer. Para el resto de las versiones de Windows, debe consultar los manuales o visitar el sitio Web de Microsoft.

## Cargar Apache para Windows

Apache para Windows se puede cargar desde la distribución de la fuente o la distribución binaria. A diferencia de la mayoría de las plataformas Unix, Windows no contiene capacidades de desarrollo como un compilador estándar de C, GCC. De modo que las únicas personas que realmente pueden compilar Apache son los desarrolladores de Windows que tienen herramientas de desarrollo como el Microsoft Visual Studio. Para la mayor parte de la gente, la distribución binaria es la única opción disponible. Aunque podemos encontrar gcc en Windows, la fuente de Apache no se compila con la versión Windows de gcc.

El sitio oficial para bajar los archivos binarios de Apache para Windows es <http://httpd.apache.org/dist/httpd/binaries/win32/>. Las distribuciones binarias se llaman httpd\_version-win32-no\_src-rnumber.msi (por ejemplo, httpd\_2\_0\_16-win32-no\_src-r1.msi). Las distribuciones de la fuente se llaman httpd\_version-win32-wint\_src.msi. Baje la última distribución binaria estándar estable.

**NOTA:** Si es un desarrollador de Windows que pretende bajar y compilar una distribución fuente de Apache, entonces puede bajar la versión comprimida de la distribución fuente en lugar de la versión .msi. Necesitará un

programa para descomprimirlo como PKUNZIP de PKWARE o WinZip para extraer los archivos de la fuente, que tiene líneas MS-DOS. No baje las distribuciones fuente que terminen en .tar.gz o .tar.Z porque son paquetes Unix con líneas para las plataformas Unix.

## Instalar binarios de Apache

Haga doble clic en el archivo httpd\_version-win32-no\_src.msi y comenzará la descarga del Microsoft Installer y la instalación de Apache. Haga clic en Next para comenzar la instalación. Le hará una serie de preguntas, que se discuten a continuación.

1. Lea y acepte la licencia que se muestra en la figura 20.1.

Debe hacer clic en la casilla "I accept the terms in the license agreement" para activar el botón Next. Haga clic en Next pasar a la pantalla siguiente.

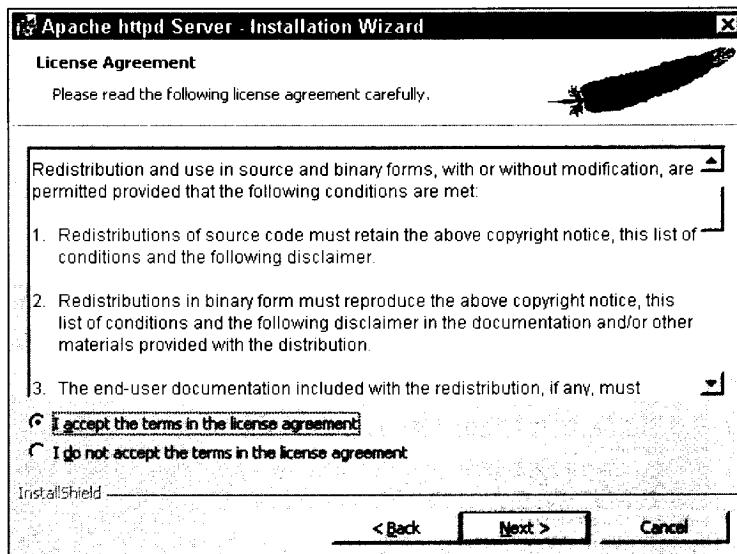


Figura 20.1. Licencia

2. Aparecerá una pantalla parecida a la que se muestra en la figura 20.2. Esta pantalla mostrará la última información (README). Navegue por esta información y cuando esté preparado, haga clic en Next para continuar.
3. La siguiente pantalla es parecida a la de la figura 20.3, y le pide que introduzca el nombre de dominio de la red, el nombre del servidor y la dirección de correo electrónico del administrador. Por defecto, el progra-

ma de instalación determina los nombre basándose en las opciones actuales de Windows. Modifique los valores por defecto por los valores adecuados. No cambie el nombre del servidor o el nombre del dominio sin tener realmente la configuración de DNS y de red apropiadas para el host. En otras palabras, no puede elegir de forma arbitraria un nombre de dominio o el nombre del host del servidor. Esta información debe ser válida para el host en el que ha instalado Apache. Haga clic en Next para continuar.

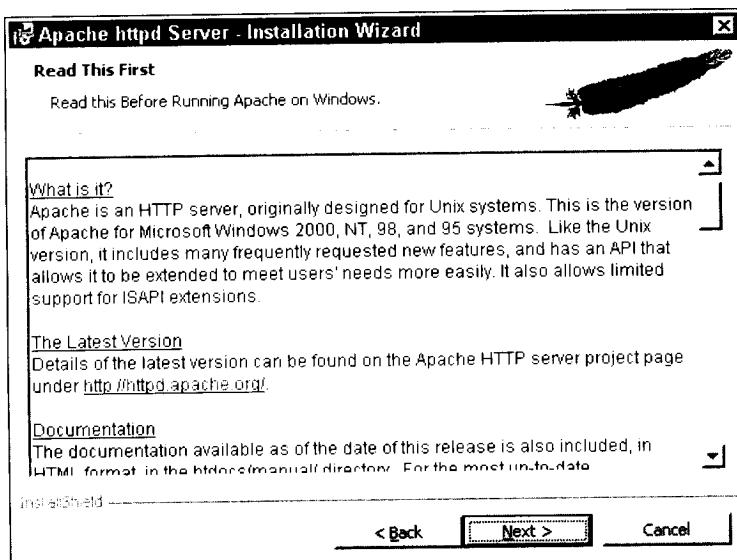


Figura 20.2. La nota Read this first

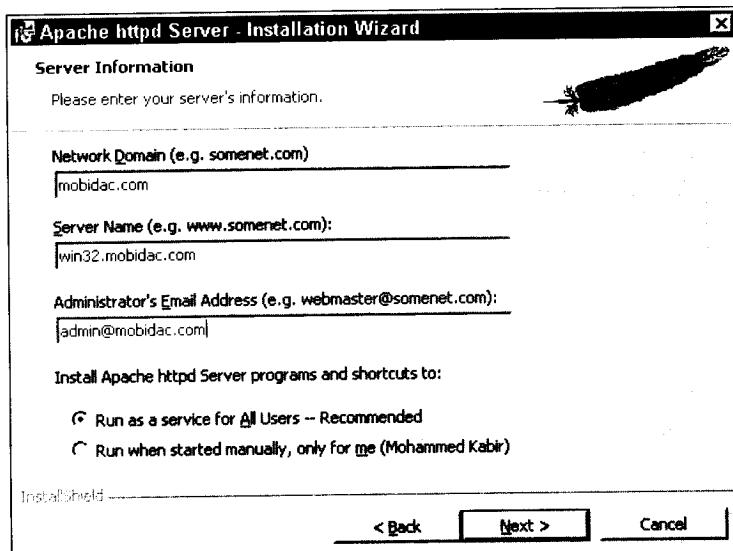


Figura 20.3. Información sobre su servidor

- Decida si quiere instalar Apache utilizando la instalación Complete (o completa, con todas las características del programa instaladas, utilizando mucho espacio en el disco) o Custom (personalizado, en el que se seleccionan los programas y archivos que se instalan), tal y como muestra en la figura 20.4. Le recomiendo que elija la instalación personalizada porque le permite saber lo que se va a instalar. Siempre tiene la opción de elegirlo todo en la instalación personalizada hasta llegar a la completa.

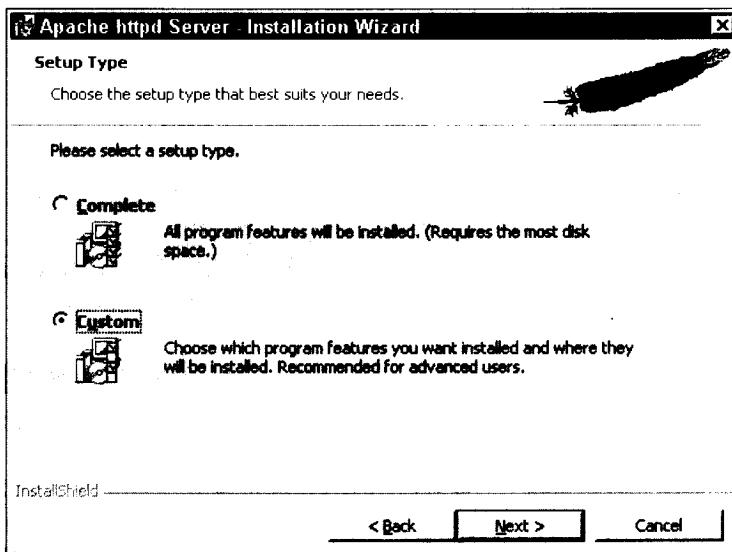


Figura 20.4. Elegir el método de configuración

- Si decide utilizar la instalación personalizada, verá una pantalla parecida a la que se muestra en la figura 20.5. Se muestran tres opciones. Puede hacer clic en cada una de las opciones para decidir si desea o no instalarla.

Por ejemplo, si hace clic en la opción Apache httpd Server, obtendrá un menú de opciones como el que se muestra en la figura 20.6.

Ha de decidir si instala la característica (httpd Apache server, en este caso) en el disco duro o en un driver de la red. Si el espacio del disco es un problema y no quiere la documentación online, puede decidir no instalar la documentación haciendo clic en esa opción para no seleccionar esa característica.

Por defecto, Apache se instala en el directorio C:\Program Files\Apache Group. Para cambiar esto, haga clic en el botón Change e introduzca un directorio distinto en la siguiente ventana de diálogo, entonces haga clic en OK para volver a esta pantalla. Haga clic en Next para continuar.

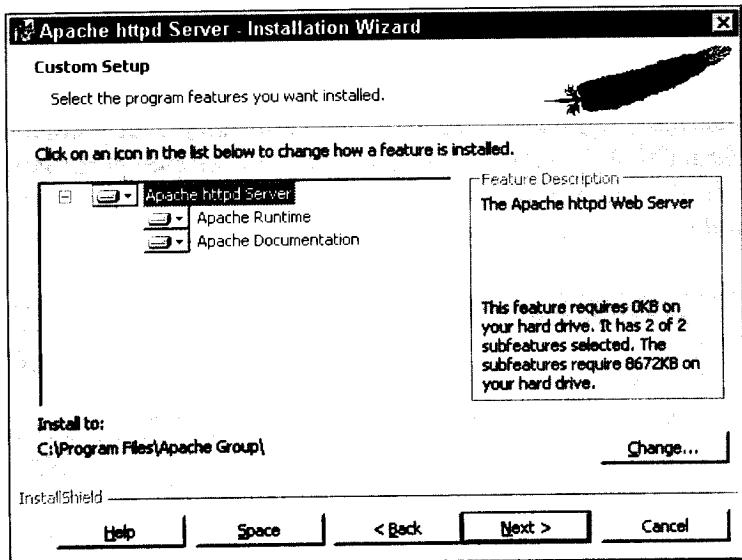


Figura 20.5. Personalizar Apache para su sitio

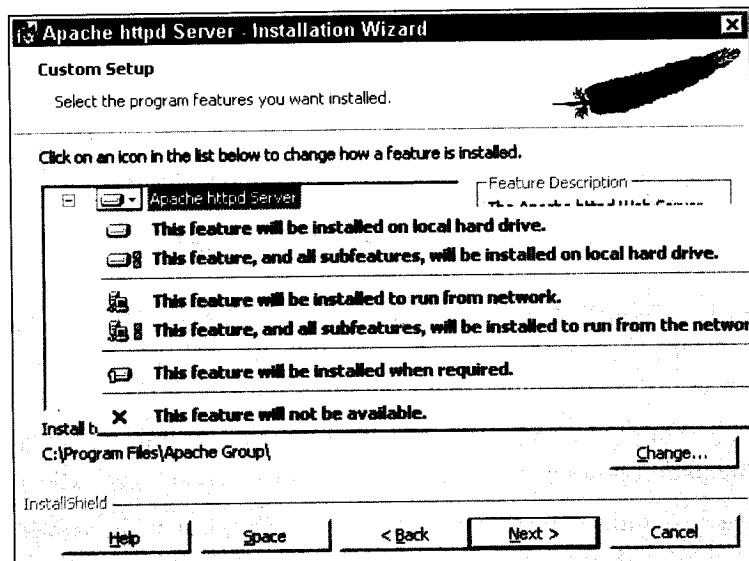


Figura 20.6. El menú de opciones de instalación

6. El programa de instalación muestra una pantalla que indica que está preparado para instalar Apache. Haga clic en Next y espere mientras se instala Apache. Cuando se completa la instalación, haga clic en Finish para que termine el programa de instalación.

Por defecto, el programa de instalación inicia Apache, por lo que debe ser capaz de ver el sitio Web por defecto de su sistema, utilizando un navegador

Web. Dirija su navegador a `http://localhost` o a `http://127.0.0.1` y verá un sitio Web parecido al que se muestra en la figura 20.7.

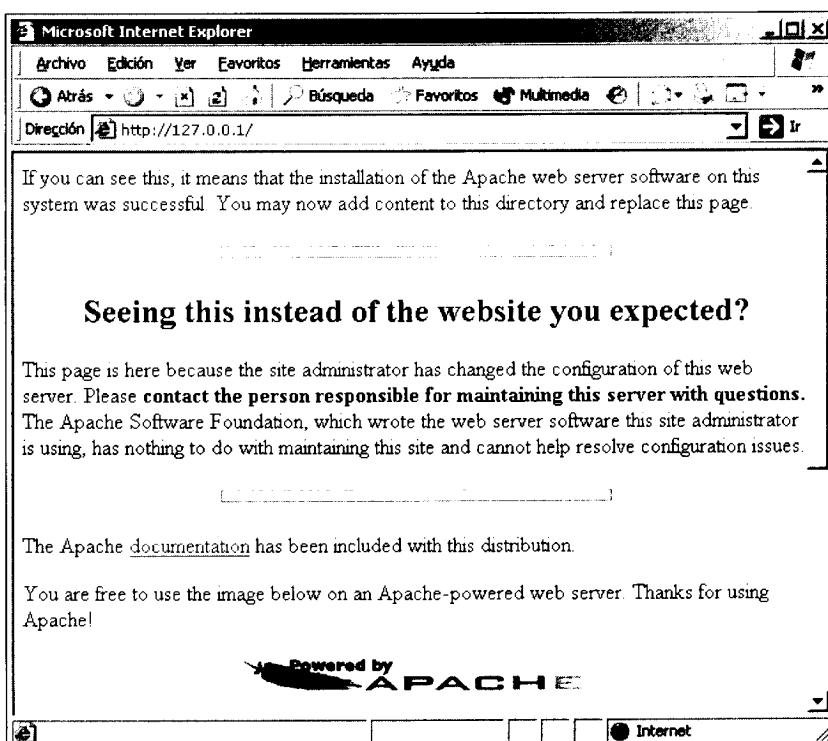


Figura 20.7. El sitio Web por defecto instalado en su sistema Windows

## Ejecutar Apache

Ejecutamos Apache bajo Windows desde la línea de comandos en una ventana de consola, o como un servicio bajo los sistemas Windows 2000 o Windows NT. En los sistemas Windows 2000 o NT, debería ejecutar Apache como un servicio autónomo. De hecho, la instalación por defecto detectará su Windows y decidirá instalarlo como un servicio para estas dos plataformas.

## Ejecutar Apache automáticamente como un servicio Windows

El programa de instalación detecta automáticamente su plataforma Windows e instala Apache como un servicio en los sistemas Windows 2000 y NT. En Windows 2000 (o en los sistemas NT), para ejecutar Apache de forma automática (o no) cada vez que se reinicia el servidor, haga lo siguiente:

- Elija Start>Settings>Control Panel>Administration Tools>Services para mostrar una pantalla parecida a la que se muestra en la figura 20.8.

**NOTA:** En los sistemas Windows 9x, no hay un buen método para ejecutar Apache como un servicio. La mejor aproximación para la plataforma 9x es ejecutar Apache en una ventana de consola.

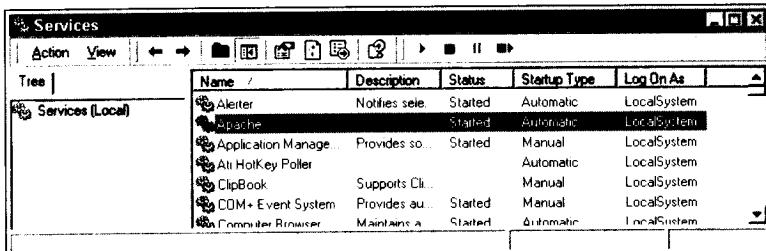


Figura 20.8. Ejecutar Apache automáticamente en el reinicio

- Haga doble clic en el nombre del servicio "Apache" y aparecerá una caja de diálogo, como la que se muestra en la figura 20.9.

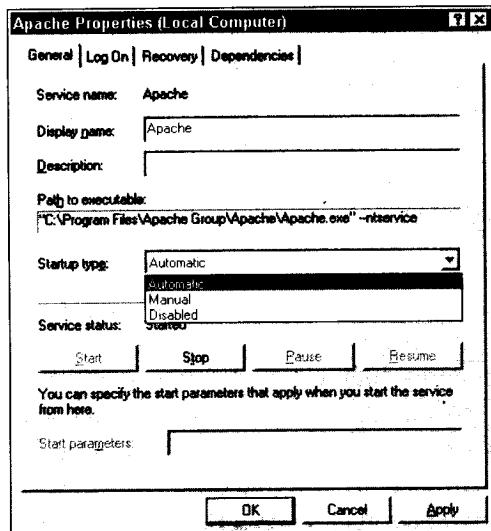


Figura 20.9. Configurar el servicio Apache

- Bajo la pestaña General verá un menú desplegable llamado Startup type. Seleccione la opción Automatic (para arranque o reinicio automático), Manual (para arranque o reinicio asistido por operador), o Disabled (para desactivarlo en el reinicio).

4. Si quiere iniciar o parar el servicio Apache, haga clic en el botón apropiado.
5. Por defecto, Apache se ejecuta utilizando la cuenta del sistema local bajo Windows 2000 o Windows NT. Si quiere cambiar esta ID de usuario a otra, haga clic en la pestaña Log On y elija la opción This account. Entonces introduzca el nombre de usuario y la contraseña adecuada. Asegúrese de que este nombre de usuario y esta contraseña son credenciales válidas de un usuario en su red Windows.

**NOTA:** Por defecto, Apache se ejecuta como la cuenta LocalSystem ("System"), que es una poderosa cuenta local que no está asociada a ningún nombre de usuario ni a ninguna contraseña. Esta cuenta no se puede utilizar para registrarse en la máquina o para conectar con otras máquinas de la misma red Windows. Si decide cambiar el usuario para el servicio Apache, asegúrese de crear una cuenta de usuario apropiada que pueda actuar como un servicio (es decir, iniciar como parte del sistema operativo), registrarse como un servicio, realizar copias de los archivos y los directorios restablecidos, tener acceso a la raíz de documentos y que tenga acceso de lectura, escritura y de eliminación de directorios de registro Apache. El nombre de usuario por defecto debería ser suficiente para la mayoría de los sistemas.

6. Bajo Windows 2000 Advanced Server, para definir una acción cuando falla el servicio Apache, seleccione la pestaña Recovery y elija las acciones apropiadas para el primer fallo, el segundo y los siguientes. Por ejemplo, puede elegir reiniciar el servicio o reiniciar la máquina automáticamente cuando falla el servicio Apache, o incluso ejecutar un programa externo para que haga algo útil como por ejemplo el papel de administrativo. La figura 20.10 muestra que cuando falla la ejecución del primer servicio Apache, se reinicia automáticamente el servicio, en el segundo fallo del servidor se reinicia la máquina automáticamente y para los siguientes fallos un programa asume el papel de administrador.
7. Cuando está configurado el servicio, haga clic en Apply para finalizar la instalación.

Puede también iniciar o parar el servicio Apache en cualquier momento desde la consola utilizando los comandos `net start apache` o `net stop apache`, respectivamente.

**NOTA:** A diferencia de muchos servicios Windows, Apache registra sus errores en su propio archivo de registro de errores. El archivo por defecto es `Logs/error.log`, que se encuentra en el directorio por defecto

C:\Program Files\Apache Groups\Apache. Solo los errores que tienen lugar durante el arranque inicial del servicio Apache, se registran en la utilidad Event Log disponible en Windows 2000 y NT.

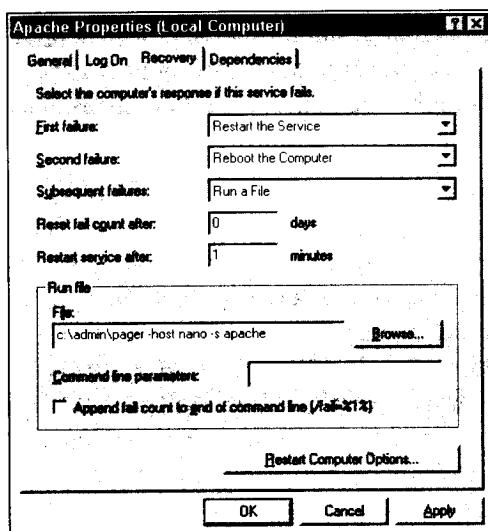


Figura 20.10. Opciones Recovery para el servicio Apache en el sistema Windows 2000 Advanced Server

## Ejecutar Apache desde el menú Start

Por defecto, el programa de instalación de Apache añadirá posibilidades en el menú Windows Start, el cual puede utilizar para iniciar, reiniciar y parar Apache del siguiente modo:

- Para iniciar Apache, elija Start>Programs-Apache httpd Server>Control Apache Server>Start.
- Para reiniciar Apache, elija Start>Programs-Apache httpd Server>Control Apache Server>Restart.
- Para parar Apache, elija Start>Programs-Apache httpd Server>Control Apache Server>stop.
- También puede editar y probar el archivo de configuración httpd.conf del siguiente modo:
- Para editar el archivo httpd.conf, elija Start>Programs-Apache httpd Server>Configure Apache Server>Edit Configuration.
- Para probar el archivo httpd.conf, elija Start>Programs-Apache httpd Server>Configure Apache Server>Test Configuration. Esto es equivalente

a ejecutar el comando `-t` de Apache desde el directorio binario de Apache. Una ventana de consola aparecerá y será capaz de ver el estado de la sintaxis de configuración.

## Gestionar Apache desde la línea de comandos

Si no sabe ejecutar Apache como un servicio, puede iniciar, parar o reiniciar Apache desde una consola (llamada habitualmente ventana DOS) utilizando los comandos siguientes.

- Para iniciar Apache, ejecute el Apache `-k start` desde el directorio binario, que es, por defecto, el `C:\Program Files\Apache Group\Apache`.
- Para reiniciar Apache, ejecute Apache `-k restart` desde el directorio binario de Apache, que es, por defecto, el `C:\Program Files\Apache Group\Apache`.
- Para parar Apache, ejecute Apache `-k stop` desde el directorio Apache, que es, por defecto, el `C:\Program Files\Apache Group\Apache`. También puede utilizar el comando Apache `-shutdown`.
- Para ejecutar Apache con un archivo de configuración distinto al archivo por defecto, especifique el nombre del archivo utilizando la opción `-f`. Por ejemplo, para ejecutar Apache utilizando `c:\test\new-httd.conf`, ejecute el comando Apache `-f "c:\test\new-httd.conf" -k start` desde el directorio que contiene los archivos binarios.

## Ejecutar varios servicios Apache

Bajo Windows, Apache normalmente ejecuta únicamente dos procesos: uno para servir todas las solicitudes HTTP utilizando varios hilos y un segundo proceso para controlar el primer proceso. Si el primer proceso muere, el segundo (el responsable de controlar el primero) reiniciará el primer proceso.

Sin embargo, puede seguir ejecutando varios servidores Apache principales, que utilicen distintos archivos `httd.conf`, desde la línea de comandos del directorio binario de Apache (por defecto es el `C:\Program Files\Apache Group\Apache`) del siguiente modo:

```
Apache -f "path_to_httd.conf"
```

Por ejemplo, puede crear dos servicios Apache principales, uno que responda al puerto 80 y otro que responda al puerto 8080, creando dos archivos `httd.conf`. Puede ejecutar cada uno de estos servicios utilizando el comando anterior. Sin embargo, en los sistemas Windows 2000 o NT, está instalado un

servicio Apache por defecto, de modo que simplemente tiene que modificar C:\Program Files\Apache Group\Apache\conf\httpd.conf para reflejar el cambio de puerto y guardarla como otro archivo. Entonces sólo tiene que ejecutar una instancia Apache utilizando el comando anterior. La configuración Apache por defecto puede seguir ejecutándose en el puerto 80.

Para crear un nuevo servicio Apache, ejecute el siguiente comando desde el directorio binario de Apache (por defecto es: C:\Program Files\Apache Group\Apache):

```
Apache -n new_service_name -f "path_to_httpd.conf" -k install
```

Ejecutando este comando, instalará un servicio nuevo llamado new\_service\_name, el cual utilizará la ruta path\_to\_httpd.conf.

También puede eliminar un servicio Apache que ya exista, utilizando el comando Apache -n existing\_service\_name -k uninstall.





# 21 Configurar Apache para Windows

---

## En este capítulo

1. Vemos las diferencias entre `httpd.conf` en Windows y `httpd.conf` en Unix.
2. Ajustamos Apache para que funcione.
3. Probamos la configuración de Apache.
4. Gestionamos Apache con Comanche.
5. Configuramos Apache para scripts CGI basados en Perl.
6. Vemos cómo se utiliza `mod_perl` en Windows.
7. Vemos cómo se utiliza PHP en Windows.
8. Vemos cómo utilizar extensiones ISAPI con Apache.
9. Vemos cómo utilizar `UserDir` en Windows.

La mayor parte de la información que encontramos en los capítulos anteriores se aplica a Apache en Windows. Sin embargo, hay ciertas diferencias entre Apache para Windows y Apache para Unix debido a las marcadas diferencias entre el

modo en el que los sistemas Unix y los sistemas Windows funcionan. Ese capítulo discute esas diferencias.

## Sintaxis httpd.conf en Windows

El archivo `httpd.conf` para un sistema Unix y un sistema Windows es distinto en cuanto al modo en el que Windows trata los nombres de rutas. Las diferencias son:

- En los sistemas Windows, los directorios y los archivos de una ruta están separados por barras inversas. Por ejemplo, `c :\temp` es una ruta válida para un directorio. Sin embargo, Apache internamente sigue utilizando una barra normal como separador, de modo que tiene que seguir utilizando este tipo de barras. Por ejemplo, `c :/temp` es correcto en el archivo `httpd.conf` pero `c :\temp` no es correcto.
- La ruta Windows suele incluir espacios en blanco. Por ejemplo, `c :\Program Files\Apache Group\Apache\htdocs` es una ruta aceptable en el mundo de Windows. Cuando utilizamos este tipo de nombres de rutas en el archivo `httpd.conf`, es necesario poner dobles comillas a la ruta completa. Por ejemplo:

```
ServerRoot "C:/Program Files/Apache Group/Apache"
```

Aquí la directiva `ServerRoot` está dirigida al directorio `C:/Program Files/Apache Group/Apache`.

## Ajustar Apache para su funcionamiento

Apache en Windows era multihilo incluso antes de Apache 2.0. Esto se debe a que los sistemas Windows como Windows 2000 y Windows NT funcionan mejor con hilos que con un manojo de procesos hijos constituyendo un servicio.

En los sistemas Windows, Apache ejecuta únicamente dos procesos. Un proceso padre controla un proceso hijo, que es responsable de servir todas las solicitudes utilizando varios hilos.

Para controlar la cantidad de solicitudes simultáneas que Apache puede servir bajo Windows, tiene que ajustar la directiva `ThreadsPerChild`. Remítase al capítulo 4 para obtener detalles sobre esta directiva. Esta directiva controla el número de hilos que se pueden crear por un solo proceso hijo Apache responsable de todas las solicitudes. El valor por defecto de 50 debería ser perfecto para la mayor parte de las solicitudes.

Si necesita una media de respuesta mayor, asegúrese de que tiene los recursos apropiados en el sistema, como RAM suficiente y una CPU rápida. De acuerdo

con el código fuente, debería ser capaz de asignar un número máximo de 4096 hilos para el módulo winnt MPM. Probablemente, un número por debajo de 256 es más apropiado para la mayoría de los sitios Web.

La directiva MaxRequestsPerChild limita el número de solicitudes que puede manejar un proceso hijo (remítase al capítulo 4 para obtener detalles sobre esta directiva). Como un solo proceso hijo procesa todas las solicitudes mediante hilos bajo Windows, esta directiva debería asignarse a un número muy alto o a 0, lo que significa que el proceso hijo nunca abandona.

## Probar la configuración de Apache

Cada vez que cambie el archivo httpd.conf, puede ejecutar el siguiente comando desde el directorio binario de Apache (C:\Program Files\Apache Group\Apache by default) para determinar si ha creado algún error de sintaxis.

```
Apache -t
```

También puede elegir la opción Start>Programs>Apache httpd Server>Configure Apache Server>Test Configuration, para producir lo mismo.

Si encuentra un error, puede arreglarlo antes de reiniciar el servidor con el comando Apache -k restart.

## Gestionar Apache con Comanche

Si usted es principalmente un usuario Windows, se preguntará dónde está el programa de administración de GUI para Apache. Lo siento, Apache no tiene. Pero hay multitud de terceras partes trabajando, e incluso las hay comerciales, que puede encontrar en <http://gui.apache.org>.

Sin embargo, Comanche (COnfiguration MANager for ApaCHE) es la que considero la mejor porque es multiplataforma, no intrusiva, extensible y muy fácil de establecer. Puede bajar Comanche de <http://www.covalent.net/projects/comanche/>. Cuando baje la distribución binaria de Comanche, extráigala y haga clic en el ícono del ejecutable para iniciar la configuración de Apache. Cuando se ejecuta el programa comanche.exe, aparece una pequeña ventana con un botón con la palabra Comanche, como la que se muestra en la figura 21.1. Haga clic en el botón con el nombre Comanche para iniciar Comanche. Debería ver una pantalla parecida a la de la figura 21.2.

Puede hacer clic en tres elementos de la izquierda para seleccionar las distintas opciones de configuración. Por ejemplo, haciendo clic en la opción Default Web Server, al final del árbol de opciones de la izquierda, mostrará una pantalla parecida a la que se muestra en la figura 21.3.

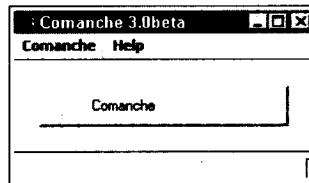


Figura 21.1. La ventana principal de Comanche

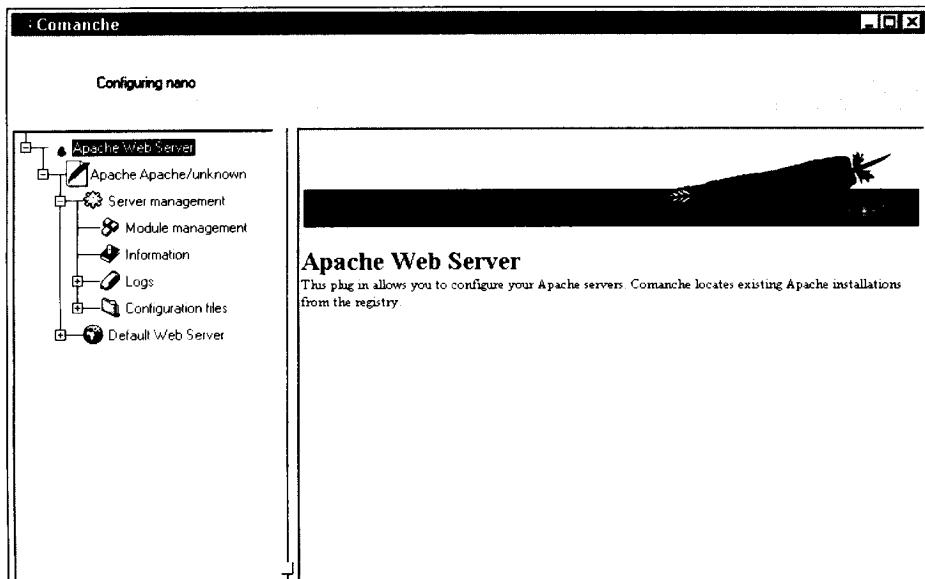


Figura 21.2. Configurar Apache con Comanche

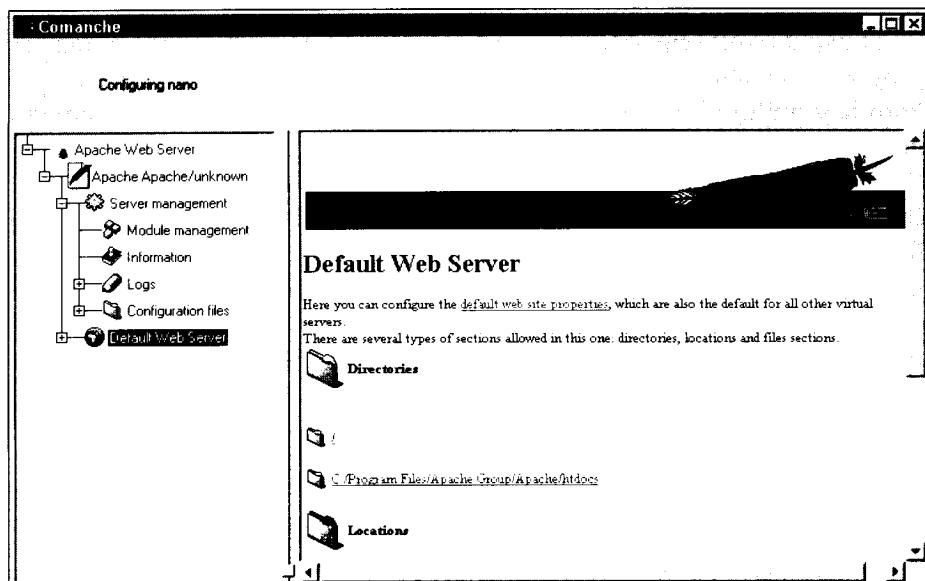


Figura 21.3. La configuración por defecto del servidor Web

Haciendo clic en el enlace de propiedades por defecto del sitio Web, obtenemos la pantalla de configuración real.

Aquí puede realizar la configuración utilizando las opciones Basic configuration (configuración básica), Module management (gestión de módulos), Server identification (identificación del servidor), Server options (opciones del servidor), Environment (entorno), Indexing (indexación), Proxy, Apache Jserv Settings (opciones para Apache Jserv), Alias o CGI Settings (opciones de CGI) del árbol del menú.

Por ejemplo, para cambiar la directiva Port a un valor distinto del valor por defecto (es decir, que no sea el 80) puede desplegar la subopción Basic Configuration y hacer clic en la opción Listening para asignarle un valor como 8080, tal y como se muestra en la figura 21.4.

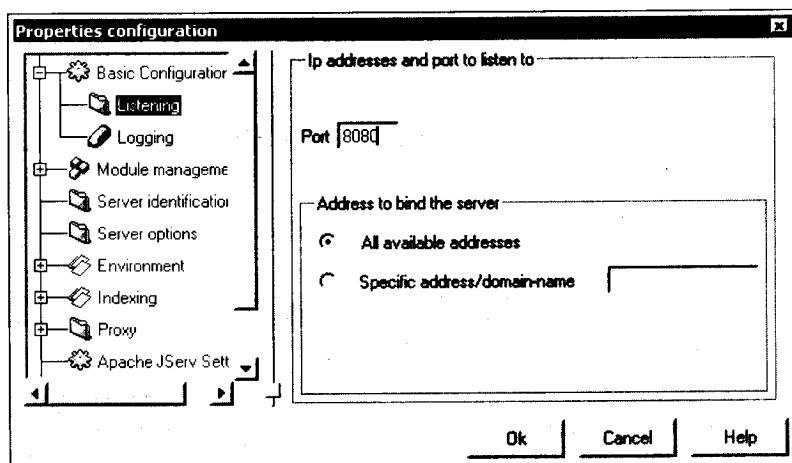


Figura 21.4. Fijar la directiva Port en 8080

Del mismo modo, para crear un alias para el script CGI llamado /cgi-bin/, apuntando al directorio "C:/Program Files/Apache Group/Apache/cgi-bin/", puede desplegar el árbol de opciones de Alias, hacer clic en la opción CGI y utilizar el botón Add para añadir el alias, tal y como se muestra en la figura 21.5. El nuevo alias creado aparece en la pantalla CGI Alias principal.

**TRUCO:** Cuando configure el servidor Web por defecto o un nuevo host virtual pueda acceder a las opciones rápidamente desplegando la opción Default Web Server de la izquierda, seleccionando y haciendo clic con el botón derecho del ratón, tal y como se muestra en la figura 21.6.

Como puede ver, una vez que navega por el árbol de opciones de la izquierda y sabe qué características se pueden crear, activar o desactivar, puede utilizar esta

herramienta para gestionar Apache desde una GUI. Sin embargo, no olvide que Comanche es un trabajo que sigue en experimentación. Utilícelo y manténgase al tanto de la aparición de nuevas versiones.

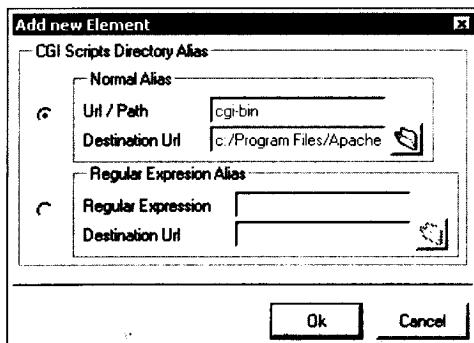


Figura 21.5. Fijar un alias para un script CGI

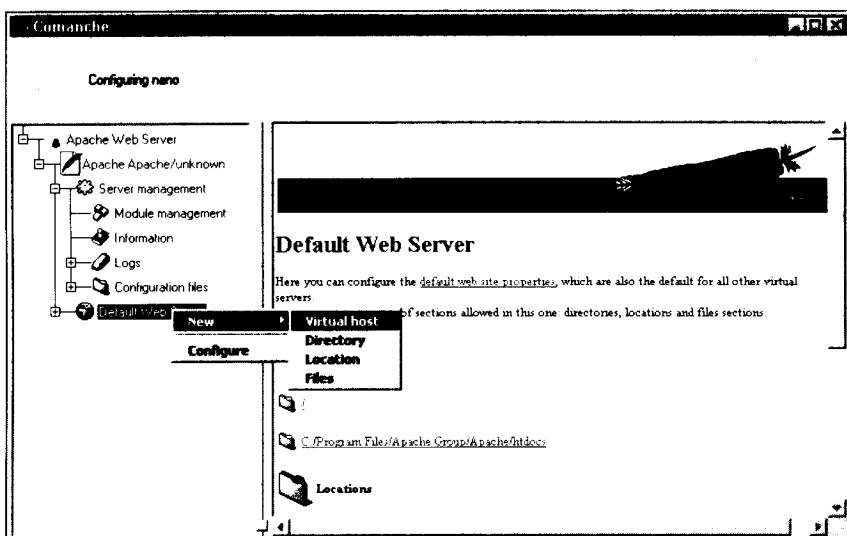


Figura 21.6. Atajo para configurar por defecto un servidor Web o un host virtual

## Configurar Apache para contenido dinámico

Al igual que en las versiones Unix de Apache, puede utilizar Apache en Windows para generar contenido dinámico utilizando scripts CGI, PHP y SSI. Sin embargo, hay algunas diferencias en la configuración para algunas de las soluciones de generación de contenido dinámico, que se discuten en esta sección. La mayoría de las directivas que se discuten aquí también se discutieron en los capítulos 4 y 5. Puede encontrar todas las opciones de sintaxis en estos capítulos.

## Ejecutar scripts CGI basados en Perl

Para ejecutar scripts CGI basados en Perl con Apache en Windows, debe instalar Perl en su sistema. Puede bajar Perl para Windows de <http://www.activestate.com>. Cuando tenga instalado Perl adecuadamente, puede configurar Apache para ejecutar scripts CGI utilizando la directiva `ScriptAlias` de la forma habitual. Normalmente, se crea un alias de script del siguiente modo:

```
ScriptAlias /cgi-bin/ "C:/Program Files/Apache Group/Apache/
cgi-bin/"
```

También puede activar scripts CGI en un directorio, utilizando la directiva `Options ExecCGI`. Ahora, para que Apache invoque el intérprete Perl apropiado, debe utilizar la siguiente línea como primera línea:

```
#!drive:/path_of_perl_interpreter
```

Esta línea se llama shebang. Por ejemplo, si ha instalado Perl en el directorio `c:\perl\bin`, puede crear scripts de Perl con la primera línea asignada en:

```
#!c:/perl/bin
```

Hay otro modo en el que puede invocar Perl para scripts CGI. Por defecto Perl registra extensiones `.pl` con el Windows Registry para ejecutarlas en el intérprete de Perl. Si asigna la directiva `ScriptInterpreterSource registry` en la parte de la configuración del servidor de `httpd.conf`, puede ejecutar cualquier script con la extensión `.pl` adecuadamente. Apache obtendrá la ruta del intérprete desde Windows Registry en lugar de la primera línea shebang.

En capítulos anteriores podrá obtener detalles de otros asuntos relacionados con CGI, que siguen siendo aplicables a Apache en una plataforma Windows.

## Ejecutar scripts mod\_perl

Para ejecutar scripts `mod_perl`, debe instalar el módulo `mod_perl`. Tendrá que instalar Perl en su sistema antes de poder instalar el módulo binario `mod_perl`, y ha de tener la última versión de Perl para Windows de <http://www.activestate.com>.

Cuando tiene Perl instalado, puede ejecutar el comando siguiente desde la línea de comandos para bajar e instalar el módulo `mod_perl`:

```
ppm install http://theoryx5.uwinnipeg.ca/ppmpackages/mod_perl-
version.ppd
```

Por ejemplo, para instalar la versión 1.25 de `mod_perl`, ejecute:

```
ppm install http://theoryx5.uwinnipeg.ca/ppmpackages/mod_perl-
1.25_1.3.19.ppd
```

Durante el proceso de instalación le pedirán que introduzca el nombre completo de la ruta, incluido el dominio, del directorio de los módulos Apache. La salida que muestra el proceso de instalación es la siguiente:

```
Which directory should mod_perl.so be placed in?  
(enter q to quit) [C:/Apache/modules]
```

Si ha instalado Apache en el directorio por defecto, el C:\Program Files\Apache Group\Apache, introduzca C:\Program Files\Apache Group\Apache\modules. El módulo mod\_perl.so se instalará en este directorio. Ahora necesita añadir la siguiente directiva en httpd.conf para bajar el módulo mod\_perl.so.

```
LoadModule perl_module modules/mod_perl.so
```

Debería asignar también la variable PATH en httpd.conf del siguiente modo:

```
PerlSetEnv PATH  
"C:\WINNT\system32;C:\WINNT;C:\perl;C:\perl\bin\"
```

Supongo que tiene instalado Windows en C:\WINNT y el directorio system32 en C:\WINNT\system32, y que ha instalado Perl en el directorio C:\perl y el intérprete Perl está almacenado en el directorio C:\perl\bin. Si alguna de estas suposiciones, que están por defecto en los sistemas Windows 2000 o NT, son incorrectas, cámbielas de forma adecuada.

**NOTA:** El directorio Windows es C:\WINDOWS (para Windows 9x) o C:\WINNT (para los sistemas Windows 2000 y Windows NT) por defecto. Si no está seguro, abra la pantalla de la consola (llamada ventana DOS) e introduzca el comando echo %windir% para desplegar el valor de la variable de entorno windir, que señala al directorio Windows OS. También puede ejecutar el comando set en el prompt para desplegar todas las variables como path, que normalmente incluye el directorio OS y el directorio system32.

Para bajar un conjunto de módulos mod\_perl al arranque, utilice un script de arranque llamado startup.pl (puede utilizar cualquier otro nombre) con la directiva siguiente:

```
PerlRequire "Drive:/your_path/startup.pl"
```

No olvide cambiar Drive:/your\_path/startup.pl con la ruta real para startup.pl. En startup.pl, cargue sus módulos utilizando las sentencias de Perl module\_name (); por ejemplo, el siguiente script startup.pl carga el módulo CGI, el módulo DBI (Database Independent Interface), y el módulo Apache::DBI que permite capturar conexiones a bases de datos:

```
#!c:/perl/bin/perl  
  
use CGI ();  
use DBI ();  
use Apache::DBI ();
```

Remítase a capítulos anteriores de este libro para obtener detalles de muchos otros asuntos relacionados con mod\_perl que se siguen aplicando a Apache en plataformas Windows. Recuerde que para cargar módulos desde CPAN tiene que utilizar el comando ppm en lugar del comando perl -MCPAN -e shell utilizado en el capítulo 16. Por ejemplo, para instalar el módulo Apache::DBI, debe ejecutar:

```
ppm install http://theoryx5.uwinnipeg.ca/ppmpackages/Apache-  
DBI.ppd
```

Del mismo modo, para instalar Apache::ASP, debe ejecutar:

```
ppm install http://theoryx5.uwinnipeg.ca/ppmpackages/Apache-  
ASP.ppd
```

## Ejecutar scripts PHP

Para ejecutar scripts PHP con Apache bajo Windows, siga estos pasos:

1. Cargue la última distribución binaria de PHP desde <http://www.php.net/distributions/>.
2. Extraiga la distribución en el directorio c:\php. Copie c:\php\php4ts.dll en el directorio system32 de Windows. Para sistemas Windows 2000 y NT, el valor por defecto es C:\WINNT\system32. Copie también el C:\php.ini-dist en el directorio Windows, que en Windows 2000 o NT es C:\WINNT por defecto.
3. php.ini funcionará para la mayor parte de los sistemas, pero si lo desea, puede editarla. Por defecto, PHP en las plataformas Windows tiene soporte integrado para MySQL y ODBC, de modo que no tiene que activar ninguna extensión. Sin embargo, debería mirar las líneas ;extension=module\_name.dll y determinar si tiene que activar alguna de estas líneas. Por ejemplo, si está pensando utilizar la librería GD (que se utiliza para generar imágenes PNG) con PHP, entonces tiene que quitar los comentarios en la línea ;extension=php\_gd.dll eliminando el punto y coma inicial.

4. Añada las líneas siguientes a httpd.conf:

```
LoadModule php4_module c:/php/sapi/php4apache.dll  
AddType application/x-httdp-php .php  
AddType application/x-httdp-php .php4
```

5. Reinicie el servidor Web Apache desde la consola utilizando el comando `net stop apache` seguido por el comando `net start apache`.
6. Sitúe un pequeño script PHP llamado `hello.php` (que se muestra a continuación) en el directorio raíz de documentos de su servidor web.

```
<html>
  <head>
    <title>Hello World from PHP</title>
  </head>
  <body>
    <?php echo "Hello World from PHP"; ?>
  </body>
</html>
```

7. Pruebe el script ejecutando el comando `http://your_server_name/hello.php`. Si ve un mensaje "Hello World from PHP" significa que ha configurado PHP adecuadamente.

A continuación puede seguir las instrucciones del capítulo 15 para crear scripts PHP para su sitio Web.

## Ejecutar extensiones ISAPI con mod\_isapi

Apache para Windows está compilado con `mod_isapi` por defecto. Este módulo le permite a Apache cargar aplicaciones basadas en Internet Server Application Programming Interface (ISAPI), que están escritas en cualquier lenguaje, de modo que se pueden ejecutar mediante Apache. ISAPI se propuso originalmente para Microsoft Internet Information Server (IIS); Apache implementa ISAPI utilizando el módulo `mod_isapi`. Sin embargo, Apache actualmente sólo permite extensiones ISAPI, por lo que, los filtros ISAPI no funcionan con Apache.

Para utilizar una extensión ISAPI (normalmente DDL, o Data Definition Language), añada la línea siguiente al archivo `httpd.conf`:

```
AddHandler isapi-isa .dll
```

Esto le permite a Apache cargar un programa de extensión ISAPI que tiene una extensión `.dll`. El módulo `mod_isapi` tiene las siguientes directivas.

### ISAPIReadAheadBuffer

La directiva `ISAPIReadAheadBuffer` asigna el máximo tamaño del buffer "Read Ahead" o de lectura de información más allá de la requerida, que está disponible para la extensión DLLs de ISAPI. Por ejemplo, cuando los datos que se van a leer superan el valor por defecto de 49152, hay que utilizar el método de retrollamada `ReadClient`.

**Sintaxis:** ISAPIReadAheadBuffer size

**Predefinido:** ISAPIReadAheadBuffer 49152

**Contexto:** configuración del servidor

## ISAPILogNotSupported

La directiva ISAPILogNotSupported activa o desactiva el registro de todas las solicitudes con características que no soportan las extensiones ISAPI. Esto únicamente resulta útil para aislar un problema relacionado con una extensión ISAPI que no funciona adecuadamente.

**Sintaxis:** ISAPILogNotSupported On | Off

**Predefinido:** ISAPILogNotSupported on

**Contexto:** configuración del servidor

## ISAPIAppendLogToErrors

La directiva ISAPIAppendLogToErrors activa o desactiva el almacenamiento de extensiones ISAPI en el registro de errores.

**Sintaxis:** ISAPIAppendLogToErrors On | Off

**Predefinido:** ISAPIAppendLogToErrors Off

**Contexto:** configuración del servidor

## ISAPIAppendLogToQuery

La directiva ISAPIAppendLogToQuery activa o desactiva el registro de datos de consulta en el registro de acceso al servidor.

**Sintaxis:** ISAPIAppendLogToQuery On | Off

**Predefinido:** ISAPIAppendLogToQuery Off

**Contexto:** configuración del servidor

# UserDir en Windows

Apache actualmente no funciona con cuentas de usuario Windows y por lo tanto no puede detectar directorios locales para los usuarios Windows. Si quiere proporcionar sitios Web para usuarios, puede utilizar el siguiente esquema:

1. Elija un directorio para el directorio local de máximo nivel para todos los usuarios. Por ejemplo, puede crear un directorio llamado C:\home y utilizarlo para este propósito.

2. Ahora puede crear un subdirectorio para cada usuario en este directorio de máximo nivel. Nombre cada uno de estos directorios utilizando cada nombre de usuario. Por ejemplo, si tiene un usuario llamado john, puede crear C:\home\john.

3. Añada la siguiente configuración a httpd.conf:

```
<IfModule mod_userdir.c>
    UserDir "C:/home/"
</IfModule>
```

4. Asigne permisos para cada directorio de usuario (C:/home/username) de modo que cada usuario de su sistema pueda leer y escribir archivos en sus respectivos directorios Web.

5. Reinicie el servidor Apache.

Ahora puede acceder a sitios Web utilizando `http://your_server_name/~username`. Por ejemplo, `http://your_server_name/~john` accederá al directorio C:/home/john.

# **Parte VI**

# **Mejorar la escalabilidad**



# 22 Apurando Apache

---

## En este capítulo

1. Utilizamos un hardware de alto rendimiento.
2. Ponemos a punto su disco duro.
3. Ponemos a punto los sistemas de archivos.
4. Compilamos e instalamos un kernel personalizado.
5. Ponemos a punto su red.
6. Ponemos a punto su configuración Apache.
7. Utilizamos el caching para aumentar la velocidad.
8. Ponemos a punto aplicaciones Web basadas en Perl.

Si nos pidiesen que escribiéramos la fórmula matemática más sencilla posible para el rendimiento, yo escribiría  $Rendimiento=f(hardware, software, red, contenido)$ , que significa que el rendimiento es función del hardware, del software, de la red y del contenido. Aunque se trata de una sentencia muy general, pone de manifiesto que el rendimiento de un servidor Web no se

puede mejorar sin poner a punto su hardware (la máquina del servidor), el software (el servidor Apache, las aplicaciones Web y el sistema operativo), la red (ancho de banda y latencia o retardo) y el contenido (tamaño y tipo).

Este capítulo discute el modo de poner a punto estos aspectos de modo que pueda mejorar el rendimiento de todo el sistema.

## Utilizar hardware de alto rendimiento

Apache se ejecuta en una gran cantidad de computadoras. Aunque la arquitectura puede variar enormemente, los componentes de hardware relacionados con los cuellos de botella en el rendimiento son prácticamente los mismos.

**NOTA:** A propósito de todo esto, como ya ha leído un capítulo de rendimiento, supongo que su sistema Apache actual se encuentra en una máquina moderna. Si está utilizando un PC clonado como servidor Web y le preocupa el tema del rendimiento, supondré que está utilizando un Pentium y no un sistema basado en i386.

### CPU

Cuando analizamos las necesidades de rendimiento de nuestro hardware, la primera pregunta que debemos hacernos es si necesitamos una CPU veloz para la computadora Apache. La respuesta depende de cómo utilicemos nuestro servidor Apache. Si el servidor sirve páginas estáticas mayoritariamente, la CPU no va a constituir un factor significativo en el rendimiento. Por el contrario, si genera una gran cantidad de contenido dinámico utilizando Server-Side Includes (SSI), scripts CGI, y similares, su servidor Apache va a necesitar un procesador rápido. En ese caso es aconsejable obtenerlo. Cuanto más rápido mejor. Debería saber, sin embargo, que una CPU rápida no va a servir de mucho a la hora de enfrentarse con un alto volumen de contenido dinámico. El candidato más probable para los cuellos de botella, en ese tipo de escenarios es la memoria RAM (que se discute en la siguiente sección).

### RAM

Siempre pensará que le falta RAM, pero una RAM cuesta dinero. La cuestión es llegar a una solución de compromiso entre la cantidad de RAM y el dinero que vamos a invertir en ella. Puede llegar a esta solución de compromiso controlando los conceptos básicos de su sistema durante la carga. Por ejemplo, en la mayoría de los sistemas Unix, puede ejecutar utilidades para controlar el rendimiento del

sistema. La figura 22.1 muestra la salida de una de estas utilidades de Unix llamada **top**.

PID	USER	PPI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	COMMAND
29164	root	18	0	1000	1060	816	R	10,7	0,2	0:01	top
1	root	0	0	532	532	468	S	0,0	0,1	0:08	init
2	root	0	0	0	0	0	SW	0,0	0,0	0:00	flushfd
3	root	0	0	0	0	0	SW	0,0	0,0	0:00	update
4	root	0	0	0	0	0	SW	0,0	0,0	0:00	rsyncd
5	root	0	0	0	0	0	SW	0,0	0,0	0:00	swapsd
6	root	-20	-20	0	0	0	SW	0,0	0,0	0:00	mdrecoveryd
45	root	0	0	0	0	0	SW	0,0	0,0	0:00	fsckd
366	root	0	0	560	560	460	S	0,0	0,1	0:00	syslogd
376	root	0	0	404	400	332	S	0,0	0,1	0:00	flpd
391	rpc	0	0	568	564	476	S	0,0	0,1	0:00	portmap
407	root	0	0	0	0	0	SW	0,0	0,0	0:00	lockd
408	root	0	0	0	0	0	SW	0,0	0,0	0:00	rpciod
418	rpcuser	0	0	724	724	612	S	0,0	0,1	0:00	rpc.statd
433	root	0	0	424	420	360	S	0,0	0,1	0:00	apmd
487	nobody	0	0	600	592	480	S	0,0	0,1	0:00	identd
494	nobody	0	0	600	592	480	S	0,0	0,1	0:00	identd
495	nobody	0	0	600	592	480	S	0,0	0,1	0:00	identd
499	nobody	0	0	600	592	480	S	0,0	0,1	0:00	identd

Figura 22.1. Salida del programa top

El programa **top** muestra una gran cantidad de información sobre la ejecución del sistema. En esta salida en concreto, la computadora está utilizando prácticamente toda su memoria física (256MB), pero aún no ha comenzado a utilizar su memoria virtual.

**NOTA:** Si está ejecutando Apache en una plataforma Windows, utilice el programa Task Manager para controlar la utilización de la memoria física, de la memoria virtual y de la CPU.

Otro programa que puede utilizar, en la mayoría de los sistemas Unix, para comprobar la utilización de la memoria virtual de su sistema, es **vmstat**. La figura 22.2 muestra una salida de una sesión de **vmstat** en la misma computadora que el ejemplo anterior. Si observa que la computadora de su servidor Web está utilizando la memoria virtual (es decir, espacio de intercambio), eso significa que tiene escasez de memoria. Es un buen momento para invertir en memoria RAM. Si no tiene la opción de comprar más memoria, o si piensa que tiene suficiente, entonces tiene que investigar algún modo de reducir la utilización de la memoria RAM de la que dispone.

## Disco duro

La siguiente pieza de hardware que tiene que considerar como un factor importante en el rendimiento de su servidor Web, es el disco duro. Un servidor Web

gasta una gran cantidad de tiempo accediendo al disco duro, y como los discos duros siguen siendo bastante lentos, suelen ser la causa principal de la perdida de rendimiento.

procs			memory				swap			io			system			cpu		
r	b	w	used	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id			
0	0	0	2680	21156	234152	85144	0	0	0	0	14	11	0	0	8			
0	0	0	2680	21156	234152	85144	0	0	0	0	107	114	1	1	98			
0	0	0	2680	21156	234152	85144	0	0	0	0	107	119	1	1	98			
0	0	0	2680	21156	234152	85144	0	0	0	0	107	114	1	1	98			
0	0	0	2680	21156	234152	85144	0	0	0	0	107	114	1	1	98			
0	0	0	2680	21156	234152	85144	0	0	0	4	140	122	1	1	98			
0	0	0	2680	21156	234152	85144	0	0	0	0	107	115	1	1	98			
0	0	0	2680	21156	234152	85144	0	0	0	0	107	119	1	1	98			
0	0	0	2680	21156	234152	85144	0	0	0	0	107	115	1	1	98			
0	0	0	2680	21156	234152	85144	0	0	0	0	107	115	1	1	98			

Figura 22.2. Salida del programa vmstat

Los discos duros normalmente son el factor limitante en el rendimiento del sistema. Por lo tanto, elegir el disco duro apropiado para su sistema es algo muy importante. Hablando en términos generales, hay diversas tecnologías de disco duro que debería considerar, como por ejemplo:

- **EIDE/IDE/ATA:** es el tipo de disco duro más habitual. Además es el tipo de disco duro más barato de los tres que vamos a ver en esta sección. También son los que dan lugar a peor rendimiento y los que, normalmente, se utilizan en un entorno local o de escritura en el que no es habitual una unidad de disco I/O masiva. Por suerte, las unidades EIDE son cada vez más veloces, ya que no estamos ante una tecnología estancada.
- **SCSI:** es el rey del mercado de servidores. Es inimaginable para mí, y para otros muchos administradores, tener un sistema servidor sin unidades de disco SCSI.
- **Disco de canal de fibra:** es la tecnología de unidades de disco más joven y atractiva, pero no se utiliza mucho debido al alto precio y la falta de conectividad. Sin embargo, las unidades de disco de canal de fibra están compartiendo mercado con SCSI. Si necesita unidades de disco de canal de fibra, debería considerar un subsistema de unidades de disco de alta calidad, como una red de sistemas de almacenamiento de datos (SAN) o un dispositivo de almacenaje.

**NOTA:** En este capítulo voy a suponer que está utilizando EIDE o SCSI.

## Entender los acrónimos

Elegir un disco duro para un sistema (de escritorio o de servidor) se convierte en algo innecesariamente difícil debido a la jerga cambiante que se utiliza en el mercado de la tecnología de unidades de disco. La tabla 22.1 define acrónimos y términos habituales para intentar ayudarle a entender las diferencias entre las distintas tecnologías que pueblan el mercado.

**Tabla 22.1.** Acrónimos y términos habituales en la tecnología de los discos duros

Acrónimo o término	Nombre estándar	Significado
IDE	ATA-1	Integrated Disk Electronics (discos electrónicos integrados).
ATA		AT Attachment (dispositivo AT). ATA es el conjunto de especificaciones IDE.
Fast-IDE o Fast-ATA	ATA-2	Segunda generación de IDE.
EIDE	ATA-3	Enhanced IDE (IDE mejorado). Proporciona soporte para las unidades de disco grandes, para varias unidades de disco (cuatro en lugar de dos), y para otras unidades de almacenaje como son las cintas y los CD-ROM.
UltraDMA/33 o UDMA/33	ATA-4	Utilizando un controlador de acceso directo de memoria (fast direct memory access, DMA), este tipo de unidad de disco, proporciona medias de transferencia intensiva de CPU menores y más rápidas.
ATAPI		ATA Packet Interface (interfaz de paquetes ATA). Es un protocolo que utilizan las cintas EIDE y las unidades de disco CD-ROM; es parecido en muchos aspectos al protocolo SCSI.

Acrónimo o término	Nombre estándar	Significado
SCSI o SCSI menor	SCSI-1	Small Computer System Interface (Interfaz de sistema para pequeñas computadoras). La implementación inicial de SCSI fue diseñada principalmente para unidades de disco pequeñas (8-bit), diferenciales, sincrónicos o asincrónicos, y estaba muy limitada con respecto al SCSI actual.
		Incluye transferencia de datos sincrónica y asincrónica a una velocidad de 5MB/segundo.
Fast SCSI o Fast-10	SCSI-2	Fast SCSI (SCSI rápido), utiliza un bus de 10 MHz en lugar del bus de 5 MHz que se utiliza en el SCSI menor. En un bus SCSI de 8-bit (menor) el bus SCSI aumenta esta velocidad teóricamente máxima desde 5MB/segundos a 10MB/segundos. En un bus de 16-bit (mayor) puede haber una media de transferencia de 20MB/segundo.
Ultra o Fast-20 SCSI	SCSI-3	Se trata de una opción de transferencia de datos sincrónica, que permite más de 20 MHz de velocidad de datos en el bus y de 40MB/segundo para el bus de 16-bit (grande), que se llama Ultra Wide SCSI.
Ultra 2 o Fast-40 SCSI	SCSI-3	Se trata de una opción de transferencia de datos sincrónica, que permite más de 40 MHz de velocidad de datos en el bus y 80MB/segundo para el bus de 16-bit (grande), que se llama Ultra2 Wide SCSI.

La mayoría de la gente utiliza discos duros IDE/EIDE o SCSI. Sólo unos pocos mantienen ambos tipos en la misma máquina, lo que no constituye ningún problema.

## Trucos en la elección de un disco duro

El rendimiento de un disco duro es un factor crítico para el servidor Web, y por tanto debería tener controladores de discos y discos SCSI de máximo nivel, en el caso de que esté pensando en ejecutar un servicio de alta calidad a demanda. La seguridad de su disco duro es también una cuestión muy importante; el fallo de un disco puede dar lugar a una gran pérdida de tiempo y a una catástrofe. A continuación tiene una serie de trucos que debería recordar a la hora de elegir un disco duro para su servidor Web.

- **Tiene que utilizar un controlador de unidades de disco SCSI de máximo nivel con un conjunto de unidades de disco SCSI de máximo nivel para su servidor Web.** Las unidades de disco SCSI ultrawide más modernas son la mejor elección. Es una buena idea utilizar varias unidades de disco en su servidor Web (de este modo, podría separar su sistema operativo de los datos Web). Utilice al menos dos unidades de disco: uno para el sistema operativo y otro para los datos. Esta posibilidad constituye, además, una buena medida de seguridad.
- **Los discos son un punto habitual de fallos.** Si le preocupan los fallos en las unidades de disco, plántese la posibilidad de realizar una copia de sus datos de forma regular. Debería obtener un sistema de array redundante de discos independientes (Array of Inexpensive Disks, RAID). Un RAID es un grupo de unidades de disco pequeñas que actúan de forma coordinada para imitar una unidad de disco grande. Si falla alguna de las unidades de disco, las otras toman el relevo hasta que se repara el fallo. La utilización de sistemas RAID podría suponer I/O de unidades de disco de alto rendimiento y una seguridad razonable para los datos de su servidor Web.
- **No utilice un disco duro lento.** Mucha gente compra sus servidores PC a fabricantes que les ofrecen paquetes en los que figuran unidades de discos IDE o EIDE de gran tamaño. Le recomiendo que evite utilizar unidades de disco IDE/EIDE en su servidor Web si quiere que su servidor Web funcione mejor que un juguete.

## Poner a punto sus discos duros EIDE/IDE (Electrónica de dispositivos integrados o Electrónica de unidades inteligentes) en Linux

Si no utiliza Linux para su servidor Apache, puede pasar a otra sección. Independientemente de su decisión respecto a la utilización de unidades de disco SCSI o IDE, debería utilizar varias unidades de disco si quiere obtener realmente un

alto rendimiento. Como mínimo, debería utilizar dos unidades de disco, uno para el sistema operativo y el software, y el otro para los datos. Para los servidores Web, recomiendo un mínimo de tres unidades de disco, utilizando el tercero para el registro generado por los sitios Web hospedados en la máquina. Mantener la I/O (Input/Output o la entrada/salida) de unidades de disco en varios dispositivos, garantiza un tiempo de espera mucho menor.

**TRUCO:** Si tiene el presupuesto suficiente para hacerlo, puede utilizar discos de canal de fibra o utilizar una red de sistemas de almacenamiento de datos (SAN). Las empresas con gran demanda de almacenamiento de datos, suelen utilizar la última opción. También puede utilizar soluciones RAID de hardware y software, que se van a discutir en el próximo capítulo.

Puede sacar mucho más rendimiento a su disco EIDE. Para empezar, tiene que determinar cuál es el rendimiento antes de los ajustes, por eso necesita alguna herramienta que pueda medir el rendimiento del estado actual de su sistema de unidades de disco.

La herramienta `hdparam` es la adecuada para realizar este trabajo, puede bajar la distribución fuente de esta herramienta de <http://metalab.unc.edu/pub/Linux/system/hardware>. La compilación y la instalación se llevan a cabo del siguiente modo:

1. Como raíz, extraiga la distribución fuente en un directorio del tipo `/usr/local/src`. Por ejemplo, yo ejecuté el comando `tar xvzf hdparm-3.9.tar.gz` en `/usr/local/src` para extraer la distribución fuente de la versión 3.9 de `hdparam`.
2. Cambie al nuevo subdirectorio creado y ejecute el comando `make install` para compilar e instalar el binario `hdparam` y la página del manual. Por defecto, el binario está instalado en el directorio `/usr/local/sbin` y se llama `hdparam`.

**ADVERTENCIA:** Como `hdparam` le permite cambiar el comportamiento de su sistema de unidades de disco IDE/EIDE, puede dar lugar, en ocasiones, a que el sistema se cuelgue como consecuencia de un uso incorrecto o de una configuración inadecuada. Le recomiendo que realice una copia de los datos antes de utilizar `hdparam`. Además, es una buena idea experimentar con `hdparam` con un solo usuario de modo que ningún otro usuario utilice el disco duro mientras esté trabajando en él. Puede reiniciar su sistema para forzarle a utilizar el modo de un solo usuario introduciendo `linux single` en el prompt `lilo` durante el inicio.

## Comprobar las opciones de su disco duro con hdparm

Una vez que ha instalado la herramienta hdparm, está preparado para investigar el rendimiento de su sistema de unidades de disco. Suponiendo que su disco duro IDE o EIDE es /dev/hda, ejecute el siguiente comando para comprobar el estado de la configuración de su disco duro:

```
hdparm /dev/hda
```

Debería ver una salida parecida a esta:

```
/dev/hda:  
multcount      =  0 (off)  
I/O support    =  0 (default 16-bit)  
unmaskirq      =  0 (off)  
using_dma       =  0 (off)  
keepsettings   =  0 (off)  
nowerr          =  0 (off)  
readonly        =  0 (off)  
readahead       =  8 (on)  
geometry        = 2494/255/63, sectors = 40079088, start = 0
```

Como puede ver, está desactivado prácticamente todo por defecto. Alguno de estos valores por defecto se tienen que cambiar para aumentar el rendimiento del sistema. Antes de continuar el proceso, necesita más información para el disco duro. Ejecute el comando siguiente:

```
hdparm -i /dev/hda
```

El comando anterior muestra la información de identificación (si existe) de unidades de discos duros que estaba disponible durante la última vez que inició el sistema. Va a necesitar esta información más tarde. El comando anterior está informando sobre el modelo, la configuración, la geometría de la unidad de disco (cilindros, cabezales, sectores), tamaño de pista, tamaño del sector, tamaño del buffer, modo soportado de DMA, modo PIO, y similares. Utilizando el siguiente comando puede comprobar el sistema de unidades de disco:

```
/dev/hda:  
Model=WDC WD205AA, FwRev=05.05B05, SerialNo=WD-WMA0W1516037  
Config={ HardSect NotMFM HdSw>15uSec SpinMotCtl Fixed DTR>5Mbs  
FmtGapReq }  
RawCHS=16383/16/63, TrkSize=57600, SectSize=600, ECCbytes=40  
BuffType=DualPortCache, BuffSize=2048kB, MaxMultSect=16,  
MultSect=16  
CurCHS=16383/16/63, CurSects=16514064, LBA=yes,  
LBAsects=40079088  
IORDY=on/off, tPIO={min:120,w/IORDY:120},  
tDMA={min:120,rec:120}  
PIO modes: pio0 pio1 pio2 pio3 pio4  
DMA modes: mdma0 mdma1 *mdma2 udma0 udma1 udma2 udma3 udma4
```

El comando está dando información sobre el modelo, la configuración la geometría de la unidad de discos (cilindros, cabezales, sectores), tamaño de pista, tamaño del sector, tamaño del buffer, modo soportado de DMA, modo PIO, etc. A continuación vamos a probar el subsistema de discos utilizando el comando:

```
/usr/local/sbin/hdparm -Tt /dev/hda
```

Verá un resultado parecido al siguiente:

```
/dev/hda:  
Timing buffer-cache reads: 128 MB in 1.01 seconds = 126.73 MB/sec  
Timing buffered disk reads: 64 MB in 17.27 seconds = 3.71 MB/sec
```

Por supuesto, sus números variarán dependiendo de su unidad de disco y del subsistema de control. Este es el estado sin ajustar de su subsistema de unidades de disco. La opción **-T** le dice a `hdparm` que compruebe el subsistema caché (es decir, la memoria, la CPU, y el caché buffer). La opción **-t** le dice al programa `hdparm` que informe sobre el estado de la unidad de disco (`/dev/hda`) leyendo datos que no se encuentren en el caché. Ejecute este comando durante algunos minutos y tome la media de MB/seg que se obtiene para su unidad de disco. Este sería en rasgos generales, el estado de rendimiento de su subsistema de unidades de disco. En este ejemplo podemos leer un rendimiento de 3.71 MB/seg, que es un número muy bajo.

## Poner a punto el modo multisector para su disco duro

Revise la salida del comando `hdparm -i /dev/hda` y busque el valor `MaxMultSect`. En este ejemplo, es 16. Recuerde que el comando `hdparm /dev/hda` mostraba un valor de 0 (off) para `multcount` en la salida. Esto significa que el modo multisector (es decir el modo bloque IDE) está desactivado.

**NOTA:** El modo multisector es una característica de la mayor parte de las unidades de disco IDE modernas. Permite a la unidad de disco transferir varios sectores por cada interrupción I/O. Está desactivada por defecto. Sin embargo, la mayoría de las unidades de disco modernas pueden realizar 2, 4, 8 o 16 transferencias de sector por cada interrupción I/O. Por eso, si asigna este modo en el valor máximo posible para su unidad de disco, que muestra el valor de `MaxMultiSect`, debería ver un aumento de la capacidad de procesamiento de un 5 a un 50 por 100 o más. Además, reducirá la sobrecarga del sistema operativo entre un 30 y un 50 por 100.

En este ejemplo el valor de `MaxMultiSect` es 16, de modo que podemos utilizar la opción **-m** de la herramienta `hdparm` para asignarlo y ver cuánto aumenta el rendimiento. Ejecute el comando siguiente:

```
/usr/local/sbin/hdparm -m16 /dev/hda
```

Si ejecutamos la prueba de rendimiento utilizando el comando hdparm -tT /dev/hda se mostrará el cambio. Para el ejemplo del sistema, el cambio es el siguiente:

```
/dev/hda:  
Timing buffer-cache reads: 128 MB in 1.01 seconds = 126.73 MB/sec  
Timing buffered disk reads: 64 MB in 16.53 seconds = 3.87 MB/sec
```

El rendimiento de la unidad de disco ha aumentado desde 3.71 MB/seg a 3.87 MB/seg. Ni mucho ni poco. Quizá su cambio sea parecido a este. Posiblemente pueda hacerlo mejor si su unidad de disco y su controlador son realmente nuevos. Probablemente alcance de unos 20 a unos 30MB/segundo. Sin embargo, sea precavido cuando curiosee con hdparm, pues se pueden dañar sus datos, por eso, tal y como se mencionó antes, tiene que realizar una copia de sus datos antes de cambiar las opciones específicas de hardware del disco duro que acabamos de ver. Si hdparm informa que el soporte I/O asignado es de 16-bit para su sistema y tiene un sistema de unidades de disco lo suficientemente nuevo (uno o dos años), debería intentar un soporte I/O de 32-bit. Puede asignarlo utilizando la opción -c en hdparm. Esta opción tiene tres valores:

0: Permite, por defecto, soporte I/O 16-bit.

1: Permite soporte de 32-bit.

3: Permite soporte de 32-bit con una secuencia de sincronización determinada que necesita muchos conjuntos de chips IDE/EIDE. Además, también es el valor que funciona en la mayoría de los sistemas.

Asigne las opciones del siguiente modo:

```
/usr/local/sbin/hdparm -m16 -c3 /dev/hda
```

Observe que se ha utilizado la opción -m16 y la opción -c3 para permitir soporte I/O de 32-bit. Ejecutar el programa con la opción -t da lugar al siguiente resultado:

```
/dev/hda:  
Timing buffered disk reads: 64 MB in 8.96 seconds = 7.14 MB/sec
```

El rendimiento del sistema de unidades de disco se ha doblado prácticamente.

## **Activar acceso directo a memoria (Direct Memory Access, DMA) en su disco duro**

Si su unidad de disco soporta acceso directo a la memoria (Direct Memory Access, DMA), podría ser capaz de utilizar la opción -d, que activa el modo DMA. Normalmente, se utilizan las opciones -d1 -X32 y -d1 -X66 juntas, para sacar partido de las capacidades DMA del sistema de unidades de disco. La primera asignación de opciones (-d1 -X32) activa el DMA modo 2 multiword para la unidad de disco, y el siguiente conjunto de opciones (-d1 -X66) permite

UltraDMA modo 2 para unidades de disco que soporten la característica UltraDMA. Estas opciones pueden aumentar enormemente el rendimiento de su unidad de disco. He visto medias de transferencia de 20 MB/seg con estas opciones en unidades de disco EIDE/ATA nuevas.

Hay otra opción, `-ul`, que puede resultar muy útil a la hora de aumentar el rendimiento general del sistema. Esta opción le permite manejar la unidad de disco para desenmascarar otras interrupciones durante el proceso de una interrupción del disco duro, lo que significa que el sistema operativo puede atender a otras interrupciones, como las de I/O de red o I/O de serie, mientras que espera que finalice la transferencia de datos basada en la unidad de disco.

Hay muchas otras opciones que puede asignar y experimentar utilizando `hdparm`; sin embargo, tiene que ser prudente con la mayoría de las opciones porque es el típico momento en el que se pueden corromper los datos. Realice siempre una copia de seguridad de los datos antes de utilizar la herramienta `hdparm`. Además, una vez que ha visto que un conjunto de opciones funciona, debería utilizarlas en el comando `hdparm` en el script `/etc/rc.d/rc.local` de modo que se asigne cada vez que se inicia el sistema:

```
hdparm -m16 -c3 -u1 -d1 -X66      /dev/hda
```

Ahora que ha ajustado su disco duro para aumentar el rendimiento, vamos a ver cómo puede poner a punto el sistema de archivos que actúa como interfaz de sus unidades de disco. Como Red Hat Linux utiliza el sistema de archivos `ext2`, vamos a ver los aspectos de la puesta a punto de `ext2` en la sección "Poner a punto el sistema de archivos `ext2` de Linux".

## Tarjeta ethernet

La última pieza de hardware, que reside en el sistema y que puede tener un impacto sobre el rendimiento, es la tarjeta adaptadora de red. Voy a suponer que su servidor Web se va a conectar a una red Ethernet. La tarjeta adaptadora de red que debería utilizar tiene que ser de alta calidad y segura. Por ejemplo, si la red Ethernet a la que quiere conectarse maneja nodos de 10Mbps o de 100Mbps, tiene que obtener una tarjeta adaptadora de 100Mbps de un fabricante con nombre de marca. El resto del hardware que puede influir en el rendimiento de su servidor se discute en la sección de redes de este capítulo.

## Poner a punto el sistema de archivos `ext2` de Linux

Si no utiliza Linux para el servidor Apache puede pasar a otra sección. Este no es el mejor sistema de archivos del mundo pero funciona razonablemente bien.

Durante años, el sistema de archivos ext2 ha sido de hecho el sistema de archivos para Linux, y por lo tanto, este es el sistema de archivos que se utiliza en este libro.

## Cambiar el tamaño del bloque del sistema de archivos ext2

Una de las formas en las que puede cambiar el rendimiento del sistema de archivos ext2 es modificando el tamaño por defecto del bloque de 1024 a un múltiplo de 1024 (normalmente menor de 4096) para servidores con archivos grandes. Para determinar qué tipo de archivos tiene en una partición ext2 determinada, haga lo siguiente:

1. Como raíz, cambie el directorio de máximo nivel de la partición ext2.
2. Ejecute los comandos siguientes, que realmente comprimen un pequeño script que utiliza las utilidades `find` y `awk`. Este script de la línea de comandos mostrará todos los archivos y sus tamaños, y finalmente le proporcionará el tamaño medio y total de la partición completa.

```
find . -type f -exec ls -l {} \; | \ 
awk 'BEGIN {tsize=0;fcnt=1;} \
{ printf("%03d File: %-06os size: %d bytes\n",fcnt++, $9,
$5); \
tsize += $5; } \
END { printf("Total size = %d\nAverage file size = %.02f\n",
\
tsize, tsize/fcnt); }'
```

3. Cuando conoce el tamaño medio del sistema de archivos, puede determinar si debería cambiar o no el tamaño del bloque. Si descubre que el tamaño medio del archivo es 8192, que es 2 x 4096, puede cambiar el tamaño del bloque a 4096.
4. No puede cambiar el tamaño del bloque de un sistema de archivos ext2 sin reconstruirlo. Por eso, tiene que realizar una copia de seguridad de todos sus archivos desde el sistema de archivos y entonces reconstruir el sistema de archivos utilizando el comando `/sbin/mke2fs /dev/partition -b 4096`. Por ejemplo, si ha realizado una copia de seguridad de la partición `/dev/hda7` y quiere cambiar el tamaño del bloque a 4096, utilice el comando `/sbin/mke2fs /dev/hda7 -b 4096`.

**NOTA:** Cambiar el tamaño del bloque a un valor superior del valor por defecto (1024) podría producir un aumento en la velocidad de lectura debido al reducido número de búsquedas y también a que la sesión `fsck` durante el reinicio podría ser más rápida, dado que la fragmentación de archivos

es mucho menor, y por otras razones. Sin embargo, incrementar el tamaño del bloque a ciegas (es decir, sin conocer el tamaño medio de los archivos) puede dar lugar a mucho espacio desaprovechado. Si el tamaño medio del archivo es de 2010 bytes en un sistema con bloques de 4096 bytes, cada archivo ocupará una media de 2086 bytes ( $4096 - 2010$ ). Por lo tanto, tiene que saber cuál es el tamaño de su archivo antes de cambiar el tamaño del bloque.

## Poner a punto el sistema de archivos ext2 con e2fsprogs

El sistema de archivos por defecto, ext2, se puede poner a punto utilizando el conjunto de programas e2fsprogs. En esta sección, veremos cómo instalar este conjunto de programas y los utilizaremos para poner a punto, comprobar rutinas de rendimiento y reparar sus discos.

### Instalar e2fsprogs

Para poner a punto el sistema de archivos ext2, tiene que instalar el paquete de utilidades e2fsprogs, del siguiente modo:

1. Cargue la distribución fuente de `e2fsprogs-version.src.rpm` (reemplace `version` con el número de la última versión) de [www.rpmfind.net](http://www.rpmfind.net). Yo bajé el paquete `e2fsprogs-1.19-0.src.rpm`. También puede obtener la fuente desde el sitio del proyecto e2fsprogs en <http://e2fsprogs.sourceforge.net>.
2. Ejecute el comando `rpm -ivh e2fsprogs-version.src.rpm` para extraer la fuente en un directorio `/usr/src/redhat/SOURCES/`. La fuente RPM lanza un archivo `e2fsprogs-version.tar.gz` que se tiene que extraer utilizando el comando `tar xvzf e2fsprogs-version.tar.gz`. Esto da lugar a un subdirectorio llamado `e2fsprogs-version`.
3. Cambie al nuevo subdirectorio.
4. Ejecute `mkdir build` para crear un nuevo subdirectorio y cambie a ese directorio.
5. Ejecute el script `../configure` para configurar el árbol fuente.
6. Ejecute la utilidad `make` para crear los binarios.
7. Ejecute el comando `make check` para garantizar que todo está correctamente construido.
8. Ejecute el comando `make install` para instalar los binarios.

Cuando tenga instaladas las utilidades e2fsprogs puede empezar utilizándolas tal y como se discute en las siguientes secciones.

## Poner a punto su sistema de archivos con tune2fs

Puede utilizar la utilidad tune2fs para poner a punto los distintos aspectos de un sistema de archivos ext2.

**ADVERTENCIA:** No debería aplicar las utilidades ext2 en un ext2 montado y debería realizar una copia de seguridad de sus datos cada vez que modifique cualquier cosa que pertenezca al sistema de archivos.

La siguiente sección discute cómo utilizar la utilidad tune2fs (parte del paquete e2fsprogs) para poner a punto un sistema de archivos ext2, que no esté montado, llamado /dev/hda7. No olvide cambiar el nombre de la partición (/dev/hda7) con el nombre apropiado. Primero vamos a ver lo que muestra tune2fs para /dev/hda7, ejecutando el comando siguiente:

```
/sbin/tune2fs -l /dev/hda7
```

La salida debería ser parecida a esta:

```
tune2fs 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95/08/09
Filesystem volume name: <none>
Last mounted on: <not available>
Filesystem UUID: 5d06c65b-dd11-4df4-9230-a10f2dd783f8
Filesystem magic number: 0xEF53
Filesystem revision #: 1 (dynamic)
Filesystem features: filetype sparse_super
Filesystem state: clean
Errors behavior: Continue
Filesystem OS type: Linux
Inode count: 1684480
Block count: 13470471
Reserved block count: 673523
Free blocks: 13225778
Free inodes: 1674469
First block: 1
Block size: 1024
Fragment size: 1024
Blocks per group: 8192
Fragments per group: 8192
Inodes per group: 1024
Inode blocks per group: 128
Last mount time: Thu Feb 15 17:51:19 2001
Last write time: Thu Feb 15 17:51:51 2001
Mount count: 1
Maximum mount count:
Last checked: Thu Feb 15 17:50:23 2001
Check interval: 15552000 (6 months)
```

Next check after:	Tue Aug 14 18:50:23 2001
Reserved blocks uid:	0 (user root)
Reserved blocks gid:	0 (group root)
First inode:	11
Inode size:	128

Los aspectos que se discuten a continuación son los que están marcados en negrita en el listado anterior:

- **Error behavior:** dicta cómo se comporta el kernel cuando se detectan errores en el sistema de archivos. Hay tres valores posibles: **continue**, **remount-ro** (sólo lectura), **panic**. La asignación por defecto es que la ejecución continúe aunque haya un error.
- **Mount count:** el tiempo que tarda en montar este sistema de archivos.
- **Maximum mount count:** significa que después de que el máximo número de modos de lectura / escritura monte el archivo, éste estará sujeto a una sesión de revisión **fsck**, durante el próximo ciclo de arranque.
- **Last checked:** muestra la fecha en la que se realizó una revisión **fsck** por última vez. El intervalo de comprobación entre dos sesiones **fsck** consecutivas.
- **Check interval:** sólo se utiliza si no se alcanza el máximo número de montajes de lectura / escritura durante el intervalo. En otras palabras, si no ha desmontado el sistema de archivos durante 6 meses, entonces, incluso aunque el número de montajes fuese sólo 2, se forzaría una revisión **fsck**, porque el sistema de archivos habría superado el intervalo de comprobación.
- **next check after:** la próxima fecha de revisión **fsck**.
- **reserved block UID** y **reserved block GID**: muestran el usuario y el grupo propietarios de la porción reservada de este sistema de archivos. Por defecto, la porción reservada es para que la utilice la raíz (UID = 0, GID = 0).

**TRUCO:** En un sistema de archivos sin montar como `/dev/hda7`, puede cambiar el número máximo de montajes de lectura / escritura a un número más adecuado para sus necesidades, utilizando la opción **-c** con `tune2fs`. Por ejemplo, `/sbin/tune2fs -c 1 /dev/hda7` forzará a que **fsck** compruebe el sistema de archivos cada vez que **fsck** basado en el tiempo. Por ejemplo, el comando `/sbin/tune2fs --i7d /dev/hda7` garantiza que se imponga la comprobación **fsck** si el sistema de archivos se vuelve a montar en modo lectura/ escritura tras una semana. Del mismo modo, el comando `/sbin/tune2fs --i0 /dev/hda7` desactiva las comprobaciones **fsck** basadas en el tiempo.

## Comprobar y reparar un sistema de archivos ext2 con e2fsck

En el caso de que tenga un sistema de archivos ext2 corrompido, puede emplear la utilidad `e2fsck` para tratar de arreglarlo. Para comprobar una partición utilizando `e2fsck`, debe desmontarla primero y ejecutar el comando `/sbin/e2fsck /dev/device` en el que `/dev/device` es su disco duro. Por ejemplo, para forzar una verificación `fsck` en un dispositivo llamado `/dev/hda7`, utilice el comando `/sbin/e2fsck -f /dev/hda7`. Esta verificación mostrará una salida como la que se muestra a continuación:

```
e2fsck 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95/08/09
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/hda7: 12/1684256 files (0.0% non-contiguous), 52897/
3367617 blocks
```

La utilidad `e2fsck` le hará una serie de preguntas, que puede obviar utilizando la opción `-p`.

## Poner a punto su sistema operativo

Apache se ejecuta en muchos sistemas operativos distintos. Debería consultar el manual de su sistema operativo para obtener los detalles de la puesta a punto y obtener un rendimiento mayor. Como la combinación más conocida es Apache en una plataforma Linux, voy a discutir cómo puede realizar el ajuste de Linux para aumentar el rendimiento. Si no utiliza Linux, debería pasar a otra sección.

Si ha instalado el vanilla kernel suministrado con Red Hat, es posible que no esté optimizado para su sistema. Normalmente, cuando instala un kernel de un fabricante proveedor, éste está fabricado para soportar la mayoría de los escenarios de instalación. Por ejemplo, un vanilla kernel podría tener soporte tanto para las unidades de disco EIDE como para las unidades de disco SCSI cuando podría necesitar únicamente soporte para SCSI o para EIDE. Por este motivo, es una buena idea empezar con un kernel de un fabricante proveedor para arrancar su sistema y tan pronto como encuentre tiempo, realizar un proceso de compilación y de instalación personalizado, que es el asunto de este capítulo.

## Compilar e instalar un kernel personalizado

El objetivo de construir un kernel personalizado es ajustar el kernel para que cubra sus necesidades. Gracias a los desarrolladores de los kernel de Linux, podemos crear un kernel personalizado en Linux. El kernel de Linux tiene un diseño

modular, lo que significa que prácticamente se puede instalar todo como módulos kernel separados. Pero volver a compilar su kernel se encuentra más allá del objetivo de este libro. Si necesita saber cómo volver a compilar su kernel, puede encontrar los detalles en <http://www.kernel.org>.

## Ajustar su sistema para aplicaciones Web en demanda

El candidato perfecto para ejecutar aplicaciones en demanda que hacen difícil la utilización de sus recursos, es probablemente un kernel de competitividad global. En mi experiencia, estas aplicaciones a menudo no son adecuadas para el kernel vanilla u otras configuraciones de recursos. Por ejemplo, un servidor de correo multihilo puede iniciar miles de archivos desde la cola de mensajes y simplemente quedarse sin manejadores de archivos o poner en marcha demasiados hilos como para alcanzar la capacidad de procesamiento simultáneo del sistema.

### Controlar el número máximo de manejadores de archivos abiertos

Un *manejador de archivos* es un recurso del sistema dirigido a un archivo abierto. Cada sistema puede tener un cierto número de manejadores de archivos abiertos. El número máximo de manejadores de archivos que pueden estar abiertos simultáneamente, a menudo controla el comportamiento de su sistema en escenarios de alta carga. En las siguientes secciones, discutiré cómo puede determinar y asignar el número máximo de manejadores de archivos para servidores Web basados en Linux.

#### Confusión entre las versiones de la distribución y del kernel

Mucha gente nueva en Linux puede sentirse confusa porque las versiones de la distribución y del kernel son distintas. Por ejemplo, a menudo me pregunto por qué hablamos de Linux 2.4 cuando el que está en el mercado es el 7.x. Gracias a distribuciones populares como Red Hat Linux, muchos recién llegados piensan que esa distribución Red Hat Linux 7.x es la última versión de Linux, debido a que por asociación todo lo que se encuentre dentro de esa distribución, también es 7.x. La realidad es que 2.4 es la última versión del kernel de Linux, y por lo tanto, cuando hablo de 2.4, me refiero al "kernel 2.4 de Linux".

#### Determinar el límite de manejadores de archivos de su sistema

Para determinar el número de manejadores de archivos que puede tener en el sistema completo, ejecute el comando `cat /proc/sys/fs/file-max`. Verá un número del tipo 4096 o 8192. Para incrementar este número de manejadores de

archivos (a menudo llamados descriptores de archivos) en nnn, simplemente añada las líneas siguientes a su script /etc/rc.d/rc.local:

```
# No olvide reemplazar nnn con el número apropiado  
echo nnn > /proc/sys/fs/file-max
```

Por ejemplo:

```
echo 10240 > /proc/sys/fs/file-max
```

Aquí los manejadores de archivos del sistema serán 10240 (10K).

### **Asignar el límite de manejadores de archivos por procesador**

Asignar el límite de manejadores de archivos por procesador en 8192, por ejemplo, garantizará que cuando un usuario se registre, podrá abrir hasta 8.192 archivos del programa bajo el control del usuario. Para asignar el límite de manejadores de archivos por proceso, tiene que seguir los pasos siguientes:

1. Edite el archivo /etc/security/limits.conf y añádale las líneas:

```
* soft    nofile 1024  
* hard    nofile 8192
```

2. Asegúrese de que su archivo /etc/pam.d/system-auth tiene una línea como la siguiente:

```
session required /lib/security/pam_limits.so
```

### **Permitir a los usuarios ejecutar menos procesos**

Para permitir a los usuarios ejecutar menos procesos, por ejemplo, 8.192 a lo sumo, añada las líneas siguientes al archivo /etc/security/limits.conf.

```
* soft    nproc 4096  
* hard    nproc 8192
```

**NOTA:** Esta asignación se aplicará a procesos y a hilos hijos que abren cada proceso. También puede configurar la cantidad de memoria que un usuario puede consumir utilizando las asignaciones de límites duros y blandos en el mismo archivo. El consumo de memoria está controlado utilizando las directivas data, memlock, rss, stack, y otras directivas parecidas. También puede controlar la utilización de la CPU de un usuario.

## **Convertir el software de su servidor Apache de competitividad global**

En el momento en el que tenga un sistema operativo de competitividad global ejecutándose en su poderoso hardware, puede empezar a investigar el software.

Para empezar tendríamos que fijarnos en el propio servidor Apache. Es posible que pueda dar lugar a un cuello de botella en el servicio Web. Seguro que sí. En el caso de que no realizase una compilación de su servidor Apache personalizada, seguro que tendrá módulos innecesarios construidos en Apache, los cuales están utilizando memoria y agotando los recursos de su sistema.

### Saber cuáles son los recursos que puede utilizar un usuario

Cada sistema tiene un número infinito de recursos disponibles del sistema. Ser capaz de controlar cuántos recursos puede utilizar un usuario es importante porque puede limitar o ampliar su capacidad de recursos en caso necesario. Para ver qué tipo de recursos del sistema puede consumir un usuario, ejecute ulimit -a (supongo que está utilizando el shell bash). A continuación se muestra un posible resultado (las líneas con el número de manejadores de archivos y el máximo número de procesos de usuarios están en negrita):

core file size (blocks)	1000000
data seg size (kbytes)	unlimited
file size (blocks)	unlimited
max locked memory (kbytes)	unlimited
<b>max memory size (kbytes)</b>	<b>unlimited</b>
open files	1024
pipe size (512 bytes)	8
stack size (kbytes)	8192
cpu time (seconds)	unlimited
<b>max user processes</b>	<b>12287</b>
virtual memory (kbytes)	unlimited

Tal y como se mencionó en el capítulo 2, hay dos formas de obtener Apache. Puede bajar la distribución binaria e instalarla en su sistema, o puede compilar su propio binario e instalarlo.

El último método es más apropiado porque le permite ajustar Apache desde el inicio. Observe los binarios que tiene actualmente y decida si necesita todo lo que le ofrece. Ejecute un comando como este:

```
httpd -l
```

que muestra todos los módulos que se han construido en su ejecutable Apache. Si observa algo que no necesita, elimínelo ejecutando el script configure con las opciones de la línea de comando apropiadas. Por ejemplo, si no pretende utilizar scripts CGI o SSI, puede eliminar estas opciones de Apache ejecutando el siguiente comando desde el directorio de distribución de la fuente:

```
./configure --prefix=/usr/local/apache \
--disable-cgi --disable-cgid \
--disable/include
```

y entonces volver a compilar y a instalar Apache con el comando `make && make install`. Si su sistema operativo le permite eliminar información simbólica innecesaria desde los ejecutables y arrancarlos, entonces puede ahorrar algo de memoria. Por ejemplo, el siguiente comando:

```
strip httpd
```

elimina símbolos de `httpd` bajo Linux. Esto disminuye un poco el tamaño del ejecutable, lo que da lugar a ahorro de RAM para cada `httpd`.

Si piensa que su ejecutable Apache (`httpd`) es tan competente como puede ser pero sigue sospechando que hay un cuello de botella en Apache, entonces examine los archivos de configuración de Apache. Algunas directivas Apache son caras en cuanto a rendimiento, y son las que, normalmente, necesitan traducciones de nombre de dominio, llamadas al sistema, manejo I/O, y manipulación de procesos. La puesta a punto de la configuración Apache se discute en la sección "Poner a punto la configuración de Apache".

## Poner a punto su red

Una vez que tiene el hardware, el sistema operativo, y el propio servidor Apache ajustados para un alto rendimiento, el siguiente punto que puede dar lugar a un cuello de botella es la propia red.

Para ajustar su red debe conocer la arquitectura de red. La mayoría de los servidores Web se ejecutan bajo una red Ethernet, ya sea la red de la empresa o la facilitada por la ISP. La forma de configurar esta red marca grandes diferencias. Por ejemplo, si tiene un servidor Apache conectado a un hub Ethernet, que está, a su vez, conectado a otros hubs que están conectados a otras terminales de trabajo o a otros servidores, tiene un buen caldo de cultivo para cuellos de botella importantes. Cada máquina puede ver todo el tráfico de la red, por eso aumenta el número de colisiones de paquetes, y es más habitual la caída de la red. Sin embargo, este tipo de cuellos de botella se remedian fácilmente utilizando un switch de red en lugar de hubs.

No se necesitá un hardware especial en los dispositivos que conectan con un switch Ethernet. La misma interfaz de red utilizada para hubs 10Base-T, funcionará con un switch Ethernet. Desde la perspectiva de estos dispositivos, conectar con un puerto de un switch es como ser el único ordenador en un segmento de red.

Un uso habitual de un switch Ethernet es romper una gran red en segmentos. Mientras sea posible acoplar un solo ordenador a cada puerto en un switch Ethernet, también será posible conectar otros dispositivos como un hub. Si su red es lo suficientemente grande para necesitar varios hubs, debería conectar cada uno de estos hubs a un puerto del switch de modo que cada hub se encuentre en un segmento distinto. Recuerde que si simplemente coloca los hubs en cascada, la red combinada es un solo segmento lógico Ethernet.

## Utilizar fast Ethernet

La red Ethernet tradicional es de 10MB/seg, lo que no es suficiente para los entornos modernos de empresa que incluyen comunicación basada en el correo electrónico, acceso a Internet, video conferencias, y otras operaciones de ancho de banda intensiva. Se necesita Ethernet de 100MB/seg. Sin embargo, 100 MB/seg o Ethernet "rápida" sigue siendo una tecnología cara si además decide utilizar los switches rápidos. Le recomiendo utilizar fast Ethernet con switches desde el primer momento. El camino de migración desde 10MB/seg a 100MB/seg puede ser caro si tiene muchos ordenadores en red. Cada ordenador de su red debe tener instalado un NIC compatible con 100MB/seg, que puede resultar caro en términos de dinero, plantilla y tiempo. Para una gran LAN con varios cientos de usuarios o más, debería modernizar un segmento cada vez. Puede empezar comprando NIC dual-speed de 10/100MB, que soportará su 10MB/seg existente, al tiempo que podrá soportar su futura infraestructura de 100MB/seg.

La utilización de fast Ethernet mano a mano con hardware de switch puede dar lugar a un alto nivel de rendimiento en su LAN. Debería considerar definitivamente esta opción. Si tiene que interconectar varios departamentos, considere una solución incluso más veloz entre departamentos: el estándar emergente Ethernet Gbit/seg es perfecto para conectar redes de áreas locales juntas para formar una WAN.

## Entender y controlar el flujo de tráfico de red

Entender cómo fluye su tráfico de red es la clave principal de la determinación de la puesta a punto de su red para aumentar el rendimiento. Observe el segmento de red que se muestra en la figura 22.3.

### Conocer las diferencias entre hubs y switches

La principal diferencia entre un hub Ethernet y un switch Ethernet es que cada puerto de un switch es un segmento lógico de red. Un ordenador conectado a un puerto en un switch Ethernet tiene un conjunto completo de anchos de banda asignados y no tiene que competir con otros ordenadores. La razón principal por la que elegimos un switch en lugar de un hub es por su capacidad para manejar direcciones. Mientras que un hub no tiene que mirar las direcciones de paquetes de datos y simplemente dirige los datos a todos los dispositivos de la red, un switch lee la dirección de cada paquete de datos y dirige correctamente el dato al recipiente o recipientes a los que está dirigido. Si el switch no leyese correctamente la dirección del paquete y dirigiese correctamente el dato, no existiría ninguna ventaja sobre un hub. La siguiente tabla muestra una lista de las principales diferencias entre un hub y un switch:

Hub Ethernet	Switch Ethernet
El ancho de banda total está limitado por la velocidad del hub, es decir, un hub 10Base-T proporciona un ancho de banda de 10MB, independientemente de los puertos que existan.	El ancho de banda total está determinado por el número de puertos en el switch. Es decir, un switch 100MB de 12 puertos, puede soportar un ancho de banda superior a 1200MB/seg, refiriéndose al ancho de banda del switch máximo que se puede agregar.
Soporta comunicaciones half-duplex, lo que limita la conexión a la velocidad del puerto, es decir, un puerto de 10MB proporciona un enlace de 10MB.	Los switches que soportan comunicaciones full-duplex ofrecen la capacidad de doblar la velocidad de cada enlace desde 100MB a 200MB.
Las reglas de cálculo de saltos (paso del origen al destino en una red), limita el número de hubs que se pueden interconectar entre dos ordenadores.	Permite a los usuarios ampliar sus redes; no hay límites en el número de switches que se pueden interconectar entre dos ordenadores.
Más baratos que los switches.	Más caros que los hubs pero mejor relación precio/ rendimiento.

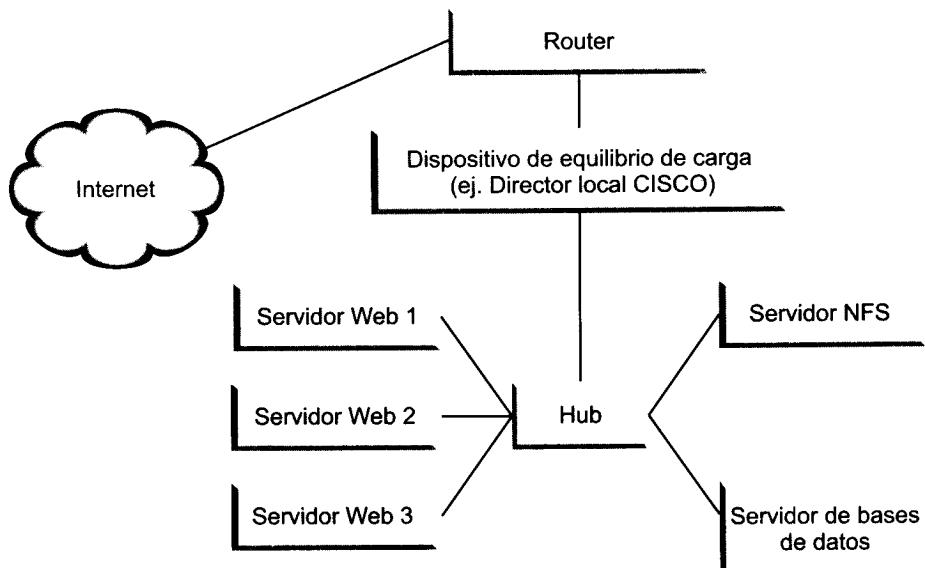
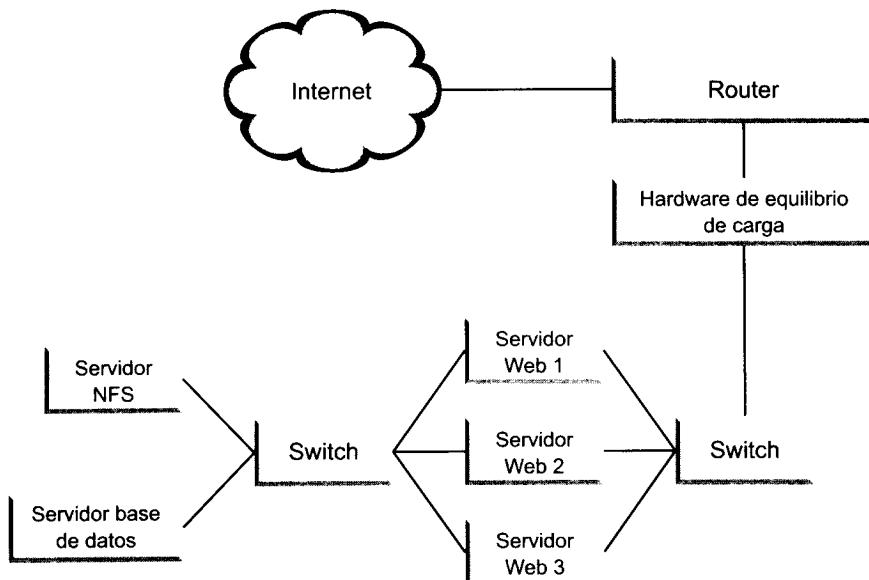


Figura 22.3. Una red Web ineficaz

Hay tres servidores Web proporcionando servicios Web a Internet y están compartiendo una red con un servidor NFS y un servidor de bases de datos. Hay varias cosas erróneas en este esquema. Para empezar, estas máquinas siguen utilizando hubs mudos en lugar de un switch. Segundo, el tráfico de NFS y de la base de datos está compitiendo con el tráfico de la Web. Si una aplicación Web necesita acceso a la base de datos en respuesta a una solicitud Web, genera una o más solicitudes de la base de datos, que, a su vez, emplea parte del ancho de banda disponible para otras solicitudes Web, ocupando innecesariamente la red o disminuyendo su capacidad de respuesta.

Para resolver este problema puede utilizar un mecanismo de control del tráfico. Primero tiene que determinar qué tráfico se puede aislar en esta red. Naturalmente, el tráfico de la base de datos y de NFS sólo se necesita para servir a los servidores Web. En este caso, el tráfico NFS y de la base de datos debería aislarse de modo que no compitan con el tráfico Web. La figura 22.4 muestra un diagrama de red modificado para la misma red.



**Figura 22.4.** Una red mejorada

Aquí, los servidores de la base de datos y de NFS están conectados a un switch que está conectado con el segundo NIC de cada servidor Web. El otro NIC de cada servidor Web está conectado a un switch que, a su vez, está conectado con el hardware de equilibrio de carga. Ahora, cuando llega una solicitud Web a un servidor Web, es servida por el servidor sin utilizar ancho de banda de otros servidores Web. El resultado es un aumento enorme de la eficacia de la red, lo que da lugar a una experiencia mucho más agradable para el usuario.

Cuando tiene un buen diseño de red, tiene que enfocar el ajuste en las aplicaciones y servicios. Dependiendo de la carga de su red, podría considerar la posi-

bilidad de desplegar varios servidores del mismo tipo para implementar un servicio más eficaz. Esto es especialmente aplicable a la Web. La siguiente sección discute el modo en el que puede emplear un esquema sencillo de equilibrio de carga utilizando un servidor DNS.

## Equilibrio de carga utilizando el servidor DNS

La idea es compartir la carga entre varios servidores de un tipo. Esto es lo que se utiliza normalmente para equilibrar el tráfico Web cuando tenemos varios servidores Web. El truco se denomina servicio de retorno al punto de origen del nombre de dominio (round-robin DNS). Suponga que tiene dos servidores Web, `www1.yourdomain.com` (192.168.1.10) y `www2.yourdomain.com` (192.168.1.20), y que quiere equilibrar la carga en `www.yourdomain.com` en estos dos servidores utilizando el truco de retorno al punto de origen. Añada las líneas siguientes a su archivo `yourdomain.com`:

```
www1 IN A 192.168.1.10  
www2 IN A 192.168.1.20
```

```
www IN CNAME www1  
www IN CNAME www2
```

Reinic peace su servidor y realice un ping a `www.yourdomain.com host`. Verá la dirección 192.168.1.10 en la salida del ping. Pare y vuelva a realizar el ping del host, y verá la segunda dirección IP, porque la configuración anterior le dice al servidor que realice un ciclo a través de los registros CNAME para `www`. En otras palabras, el host `www.yourdomain.com` es `www1.yourdomain.com` y `www2.yourdomain.com`.

Ahora, cuando alguien introduce `www.yourdomain.com`, el servidor saca la primera dirección una vez, y después saca la segunda para la siguiente solicitud, manteniendo un ciclo entre estas dos direcciones.

**ADVERTENCIA:** Una desventaja del truco de retorno al punto de origen es que el servidor no puede saber qué sistema está muy cargado y cuál no (son ciclos ciegos). Si uno de los servidores se estropea o se vuelve inaccesible por alguna razón, este sistema vuelve a la dirección IP del servidor estropeado de forma normal. Esto puede resultar caótico, porque algunas personas serán capaces de acceder al sitio y otras no.

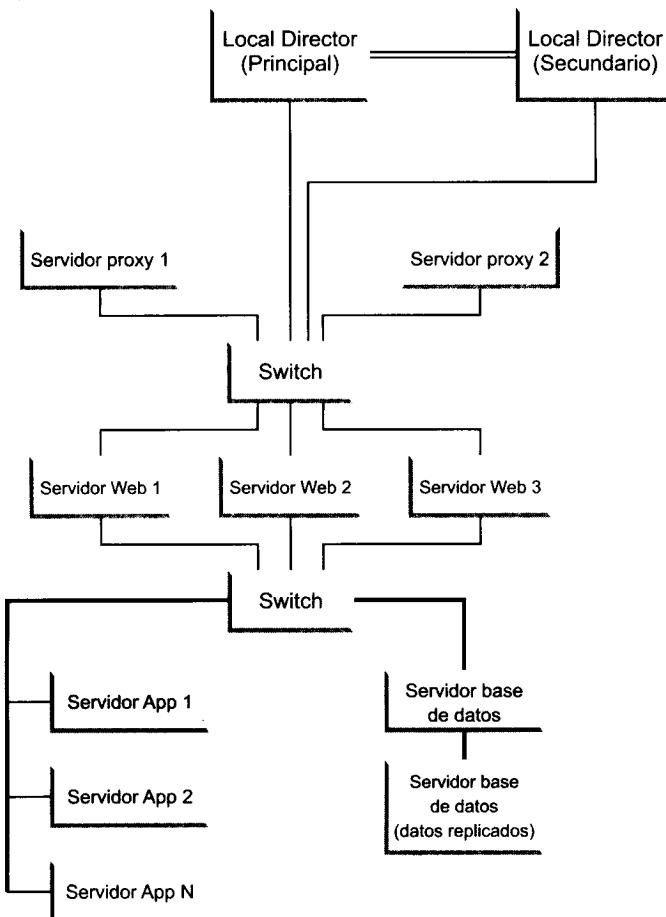
## Utilizar hardware de equilibrio de carga

Si su carga le demanda un equilibrio de carga más inteligente y es importante la comprobación del estado de su servidor, lo mejor que puede hacer es obtener

una solución de hardware que utilice los nuevos productos Director, como Web Director ([www.radware.com](http://www.radware.com)), Ace Director ([www.alteon.com](http://www.alteon.com)) o Local Director ([www.cisco.com](http://www.cisco.com)).

La figura 22.5 muestra una red Web, que consiste en dos Local Directors CISCO; un conjunto de servidores proxy; servidores Web Apache; mod\_perl, PHP, servidores de aplicaciones Java Servlet; y servidores de bases de datos.

Todos los dominios Web alojados en esta red, tienen una dirección IP virtual que se traduce en el Local Director. El Local Director utiliza esta información y la información del estado del servidor, que se almacena, para determinar de dónde obtener el contenido. El segundo Local Director simplemente funciona a la espera en caso de que falle el primero. Local Director le permite una recuperación de estado cuando se conectan dos Local Directors mediante un cable especial. Si el primero falla, entonces el segundo puede tomar el poder sin interaccionar con el exterior.



**Figura 22.5.** Una red Web que utiliza Local Director para equilibrio de carga

**TRUCO:** Si realmente le preocupa la seguridad de su red Web, no puede tener un solo punto de fallo. Por ejemplo, si utiliza un servidor de bases de datos, asegúrese de tener otro que esté replicando los datos en tiempo real de modo que pueda recuperarlos en caso de fallo del servidor.

## Poner a punto la configuración de Apache

Una vez que tiene configurado el hardware, el sistema operativo y la red, y tiene instalado el software de Apache (ya hemos discutido estos procesos en este capítulo), está preparado para poner a punto la configuración de Apache. Las secciones siguientes tratan varias opciones que puede aplicar fácilmente para aumentar el rendimiento del sistema.

### Minimizar las búsquedas DNS

Si la directiva `HostnameLookups` tiene el valor `On`, Apache realizará búsquedas DNS para cada solicitud para traducir la dirección IP en un nombre de host. Esto puede disminuir enormemente el rendimiento del servidor. Por eso, debería considerar seriamente no utilizar búsquedas de host para cada solicitud. Asígnele a `HostnameLookups` el valor `Off` en `httpd.conf`.

Si tiene que traducir direcciones IP en nombres de host con propósitos de procesamiento de registros, utilice la herramienta `logresolve`. Ver el capítulo 8 para obtener detalles.

### Apurar el servicio de archivos estáticos

Aunque todo el mundo reclama contenido Web dinámico que sea manejado por bases de datos o servido por servidores de aplicaciones de lujo, las páginas Web estáticas siguen existiendo. De hecho, el contenido dinámico no tiene probabilidades de reemplazar por completo a las páginas Web estáticas en un futuro cercano, ya que servir páginas estáticas es normalmente más rápido que servir contenido dinámico. Algunos sistemas de contenido dinámico incluso crean páginas estáticas dinámica y periódicamente como contenido caché para una entrega más rápida. Esta sección discute el modo de aumentar la velocidad de la distribución de páginas estáticas utilizando Apache y el módulo HTTP kernel de Linux.

#### Reducir el manejo de I/O para entregar páginas estáticas rápidamente

Cuando se realiza una solicitud a Apache de una página Web estática, éste realiza una búsqueda en el árbol de directorios de archivos `.taces` para garan-

tizar que la página solicitada se puede enviar al navegador Web. Por ejemplo, si un servidor Apache que se está ejecutando en `www.nitec.com`, recibe una solicitud del tipo `http://www.nitec.com/training/linux/sysad/intro.html`, Apache realiza las comprobaciones siguientes:

```
/.taces  
%DocRoot%.taces  
%DocRoot%/training/.taces  
%DocRoot%/training/linux/.taces  
%DocRoot%/training/linux/sysad/.taces
```

`%DocRoot%` es el directorio raíz de documentos asignado por la directiva `DocumentRoot` en el archivo `httpd.conf`. De modo que si este directorio es `/www/nitec/htdocs`, entonces se realizan las siguientes comprobaciones:

```
/.taces  
/www/.taces  
/www/nitec/.taces  
/www/nitec/htdocs/.taces  
/www/nitec/htdocs/training/.taces  
/www/nitec/htdocs/training/linux/.taces  
/www/nitec/htdocs/training/linux/sysad/.taces
```

Apache busca el archivo `.taces` en cada directorio de la ruta traducida (desde la URL solicitada) del archivo solicitado (`intro.html`). Como puede ver, una URL que solicita un solo archivo puede dar lugar a varias solicitudes de manejo I/O para leer varios archivos. Esto puede agotar los sitios de alto volumen. En estos casos, su mejor opción es desactivar todas las comprobaciones de archivos `.taces`.

Por ejemplo, las siguientes directivas de configuración, que están situadas dentro de la sección del servidor principal (que no está dentro de la directiva `VirtualHost`) del archivo `httpd.conf`, desactivarán la comprobación de `.taces` para cada solicitud URL.

```
<Directory />  
    AllowOverride None  
</Directory>
```

Cuando se utiliza la configuración anterior, Apache simplemente realizará un solo manejo I/O para leer el archivo estático solicitado y por lo tanto obtendrá rendimiento en los escenarios de alto volumen de accesos.

## **Reducir las llamadas al sistema y los manejos I/O para los archivos simbólicos**

En los sistemas Unix y en los sistemas de tipo Unix que ejecutan Apache, los enlaces simbólicos presentan un peligro. Si utilizamos un enlace simbólico mal colocado, un usuario Web puede ver archivos y directorios que no deberían estar

disponibles mediante la Web. Por ese motivo Apache ofrece un modo de desactivar enlaces simbólicos o únicamente seguir un enlace simbólico si la ID del usuario del enlace simbólico, coincide con el dueño del servidor. Por ejemplo, la configuración siguiente de la sección del servidor principal (es decir, fuera de cualquier configuración de un host virtual) de `httpd.conf` le dirá a Apache que no siga los enlaces simbólicos, desactivando todos los accesos a enlaces simbólicos mediante la Web.

```
<Directory />
    Options -FollowSymLinks
</Directory>
```

Esto tiene un precio significativo en el rendimiento. Para cada solicitud, Apache realiza una llamada condicional al sistema, `lstat()`, para garantizar que no se está violando la política de no seguir enlaces simbólicos.

Para aumentar el rendimiento al tiempo que tenemos enlaces simbólicos y seguridad, realice lo siguiente:

1. Encuentre la manera de no utilizar ningún enlace simbólico en el árbol de documentos Web. Puede utilizar el comando `your_top_web_directory -type l -print` para encontrar todos los enlaces simbólicos que existan en su directorio Web de máximo nivel, entonces puede determinar cómo evitarlos.
2. Utilice la configuración siguiente en la sección del servidor principal de `httpd.conf` para activar enlaces simbólicos:

```
<Directory />
    Options FollowSymLinks
</Directory>
```

3. Si tiene que desactivar enlaces simbólicos, considere estrechar el objetivo del directorio con un nombre de directorio concreto. Por ejemplo, si quiere desactivar enlaces simbólicos en un directorio llamado `my_dir`, pero permitir enlaces simbólicos en el resto (por motivos de rendimiento), puede utilizar esta configuración:

```
<Directory />
    Options FollowSymLinks
</Directory>
```

```
<Directory /my_dir>
    Options -FollowSymLinks
</Directory>
```

4. También puede utilizar la directiva `SymLinksIfOwnerMatch`:

```
<Directory />
    Options FollowSymLinks
</Directory>
```

```
<Directory /my_dir>
    Options -FollowSymLinks +SymLinksIfOwnerMatch
</Directory>
```

Aquí Apache seguirá enlaces simbólicos en el directorio `/my_dir` si el ID del dueño coincide con el ID del usuario del servidor.

## Poner a punto su configuración utilizando ApacheBench

El servidor Apache contiene una herramienta llamada ApacheBench (ab), que está instalada por defecto en el directorio `bin` del directorio de instalación de Apache. Utilizando esta ingeniosa herramienta, puede poner a punto la configuración de su servidor.

Dependiendo de la elección que realice de su módulo multiproceso (MPM, prefork, threaded, perchild), tendrá que ajustar los valores de la siguiente configuración por defecto:

```
<IfModule prefork.c>
    StartServers          5
    MinSpareServers       5
    MaxSpareServers      10
    MaxClients           20
    MaxRequestsPerChild  0
</IfModule>

<IfModule threaded.c>
    StartServers          3
    MaxClients            8
    MinSpareThreads       5
    MaxSpareThreads      10
    ThreadsPerChild       25
    MaxRequestsPerChild   0
</IfModule>

<IfModule perchild.c>
    NumServers            5
    StartThreads          5
    MinSpareThreads       5
    MaxSpareThreads      10
    MaxThreadsPerChild    20
    MaxRequestsPerChild   0
</IfModule>
```

Ajustar estas directivas al azar no es una buena idea. Como su sitio Web, su patrón de tráfico y sus aplicaciones son probablemente distintos a las de otros sitios, no hay una fórmula totalmente adecuada para calcular los valores apropiados para estas directivas.

Le mostraré una técnica, sin embargo, que utiliza ApacheBench para determinar los valores apropiados.

**ADVERTENCIA:** Debería utilizar la herramienta ApacheBench en un sistema (o en varios sistemas) distinto del propio servidor Web, porque intentar acotar en el mismo servidor utilizando un modelo cliente/ servidor le dará una información falsa. La herramienta benchmark, ab, en sí misma resta recursos del servidor y falsea sus resultados. Por lo tanto, debe ejecutar ab en una máquina distinta. Le recomiendo ejecutar ab en varias máquinas para simular mejor las cargas.

**NOTA:** Tendrá que compilar Apache en otras máquinas para obtener el binario ab instalado en un sistema servidor que no se Web. Puede instalar un RPM binario de Apache en este tipo de sistemas y desinstalarlo una vez que termina la puesta a punto. Ver los capítulos 2 y 3 para obtener detalles sobre el modo de instalar y configurar Apache.

Determine un objetivo para su servidor. Realice una estimación (o adivinación) de cuántas solicitudes quiere ser capaz de servir desde su servidor Web. Escríbalo en una frase objetivo del tipo, "Deseo servir N solicitudes por segundo." Reinicie su servidor Web y desde un sistema distinto del servidor Web, ejecute el comando ab del siguiente modo:

```
./ab -n number_of_total_requests \
      -c number_of_simultaneous_requests \
      http://your_web_server/page
```

Por ejemplo:

```
./ab -n 1000 -c 50 http://www.domain.com/
```

La herramienta ApacheBench realizará 50 solicitudes concurrentes y un total de 1.000 solicitudes. A continuación se muestra un posible resultado:

Server Software:	Apache/2.0.16
Server Hostname:	localhost
Server Port:	80
Document Path:	/
Document Length:	1311 bytes
Concurrency Level:	50
Time taken for tests:	8.794 seconds
Complete requests:	1000
Failed requests:	0
Total transferred:	1754000 bytes
HTML transferred:	1311000 bytes
Requests per second:	113.71
Transfer rate:	199.45 kb/s received

Connnection Times (ms)			
	min	avg	max
Connect:	0	0	5
Processing:	111	427	550
Total:	111	427	555

Observe que Requests per second (solicitudes por segundo) es 113.71 para el acceso a la página local del sitio <http://www.domain.com>. Cambie el contador de solicitudes concurrentes a un número mayor y observe cómo maneja el servidor la carga concurrente adicional.

Ahora cambie los valores de MaxClients, ThreadsPerChild, MaxThreadsPerChild, y similares, basándose en su MPM, reinicie Apache, y aplique la misma prueba de acotación utilizando ab como antes. Debería ver su Requests per second (Solicitudes por segundo) crecer y bajar dependiendo del número que esté intentando. A medida que va probando números cambiando los valores de las directivas, asegúrese de registrar los valores y el rendimiento de modo que pueda determinar cuál es el mejor valor en su caso.

## Utilizar el caching para aumentar la velocidad

Utilizar el caching para el contenido Web no es un concepto nuevo. Los sitios Web más ocupados, implementan el caching utilizando servidores proxy o algún otro mecanismo.

Aquí veremos dos opciones entre las que podrá elegir. Además, debería investigar las capacidades proxy de Apache utilizando el módulo mod\_proxy discutido en el capítulo 10.

## Meter los archivos muy utilizados en la memoria caché con mod\_fcache

El módulo mod\_fcache para Apache introduce un tipo determinado de archivos en la memoria caché. Los archivos que están en la memoria caché se almacenan en el espacio de memoria del servidor principal y son accesibles a todos los procesos hijo de Apache. Puede bajar este módulo de [www.fractal.net/mod\\_fcache.tgz](http://www.fractal.net/mod_fcache.tgz). Para compilar e instalar este módulo, siga los siguientes pasos:

1. Como raíz, extraiga la fuente del módulo utilizando el comando tar xvzf mod\_fcache.tar.gz y copie el nuevo directorio creado en el subdirectorio de módulos de la distribución fuente de Apache. Por ejemplo, si ha instalado la fuente Apache en /usr/local/src/httpd\_2.0.16 y fcache en /usr/local/src/fcache, entonces

ces puede copiar los archivos del módulo utilizando el comando `cp -r /usr/local/src/fcache /usr/local/src/httpd_2.0.16/modules`.

2. Cambie el directorio al subdirectorio `modules/fcache` de la distribución fuente de Apache. Observe el archivo `config.m4` y decida si necesita cambiar algo para su sistema. Lo más probable es que no tenga que realizar ningún cambio. Si cambia algo, debe tener claro en qué punto se encuentra.
3. Ejecute `autoconf`. para configurarlo todo.
4. Vuelva a cambiar el directorio a la distribución fuente de Apache de máximo nivel y ejecute el script `configure` de Apache con todas las opciones que utilice normalmente (ver el archivo `config.status`) y la opción `-enable-fcache`.
5. Compile e instale Apache como siempre, utilizando el comando `make && make install`.
6. Reinicie el servidor Web Apache utilizando el comando `/usr/local/httpd/apachectl restart`.

Ahora está preparado para utilizar este módulo. Para meter en la memoria caché las imágenes GIF que se sirven desde un directorio llamado `common_images`, por ejemplo, utilizamos el siguiente segmento de configuración en `httpd.conf`:

```
<Directory /common_images>
    fcache          On
    fcache_CacheTypes   image/gif
    fcache_MaxSize     10240
    fcache_RecomputeTables 600
</Directory>
```

Algunos puntos a tener en cuenta sobre el segmento anterior:

- `fcache` activa el caching de módulos.
- La directiva `fcache_CacheTypes` asigna el tipo MIME para el caching. La configuración del ejemplo lo asigna a `image/gif`. Si quiere meter en la memoria todos los tipos de imágenes, puede utilizar `image/*`.
- `fcache_MaxSize` asigna el tamaño del caché. Aquí la memoria caché es de 10MB (1024KB x10). Recuerde que debe tener una gran cantidad de memoria para cachear archivos.
- La directiva `fcache_RecomputeTables` determina el tiempo en segundos para recalcular las tablas de caché. El valor por defecto de 10 minutos es suficiente para la mayoría de los objetivos.

**TRUCO:** Para ver la estadística de caché, puede crear la siguiente configuración:

```
<Location /fcache-stats>
  SetHandler fcache-stats-handler
</Location>
y entonces acudir a la página http://your_web_server/fcache-
stats.
```

## Adquirir habilidad con el servidor proxy-caché Squid

Squid es un servidor proxy de caching compatible con HTTP 1.1, de código fuente abierto, que puede utilizar para mejorar su experiencia de navegación en la Web. Puede bajar la última distribución fuente estable de Squid de [www.squid-cache.org](http://www.squid-cache.org).

De forma ideal, tendría que ejecutar el servidor proxy-caching con dos interfaces de red. Una interfaz lo conecta con el gateway de Internet o con el router y el otro lo conecta a la red interna.

**TRUCO:** Desactivar el redireccionamiento de IP en el sistema proxy-caché garantiza que nadie puede saltarse el servidor proxy y acceder directamente a Internet.

Las siguientes secciones discuten la instalación y la configuración de Squid.

### Compilar e instalar el servidor proxy-caché Squid

Para compilar e instalar Squid, siga los pasos siguientes:

1. Como raíz, extraiga la distribución fuente utilizando el tar xvzf squid-version.tar.gz (donde version es el número de la última versión del software Squid).
2. Ejecute el comando ./configure --prefix=/usr/local/squid para configurar el código fuente de Squid para su sistema.
3. Ejecute make all; make install para instalar Squid en el directorio /usr/local/squid.

Una vez que tiene instalado Squid, tiene que configurarlo (ver la siguiente sección).

## Configurar Squid

Para configurar Squid, siga los siguientes pasos:

1. Tiene que crear un grupo llamado nogroup empleando el comando groupadd nogroup. Squid utilizará este grupo.
2. Ejecute el comando chown -R nobody:nogroup /usr/local/squid para asignar el dueño del directorio /usr/local/squid y todos sus subdirectorios al usuario nobody y al grupo llamado nogroup. Esto le permite a Squid (ejecutándose como el usuario nobody) crear directorios y archivos caché y escribir registros. Modifique el archivo /usr/local/squid/etc/squid.conf tal y como se discute en los pasos siguientes.
3. Decida en qué puerto quiere ejecutar el proxy-caché. Yo he utilizado el 8080, que es el que utilizan la mayoría de los proxy-caché. Añada la línea siguiente en squid.conf:

```
http_port 8080
```

Esto le dice a Squid que escuche en el puerto 8080 las solicitudes proxy.

**TRUCO:** Puede utilizar otro puerto, si así lo prefiere. Asegúrese de no utilizar un puerto que ya se esté utilizando en otro servidor. En teoría, tiene que utilizar números de puerto por encima de 1024 para evitar colisionar con servicios estándar, pero si sabe que no está ejecutando un servidor Web en el puerto 80 y quiere ejecutar su proxy-caché en ese puerto, puede hacerlo. Además, una forma rápida de comprobar si está disponible un puerto es ejecutar el comando telnet localhost portnumber donde portnumber es el número de puerto que quiere utilizar para su proxy-caché. Si obtiene un mensaje de fallo de conexión, el puerto no se está utilizando.

4. Tiene que definir el lugar en el que quiere guardar los datos cacheados. Defina las siguientes líneas en el squid.conf:  

```
cache_dir ufs /usr/local/squid/cache 100 16 256
```

Esto le dice a Squid que quiere almacenar los datos cacheados en /usr/local/squid/cache. Si tiene una base de usuarios muy grande que utilizará este proxy-caché, es una buena idea tener varios directorios repartidos en distintas unidades de disco. Esto reduce la espera relacionada con I/O, ya que varias unidades de disco independientes son siempre más rápidas que una sola unidad de disco.
5. La configuración de Squid por defecto no permite ninguna conexión desde ningún sitio; esta es una característica de seguridad que se suele llamar

"rechaza a todo el mundo, sólo permite el acceso a los que deberían tenerlo." Por eso, tiene que crear una lista de control de acceso (ACL) que permita a su red acceder al proxy-caché. Por ejemplo, si su dirección de red es 192.168.1.0 con una máscara de red 255.255.255.0, entonces puede definir la siguiente línea en squid.conf para crear una ACL para su red:

```
acl local_net src 192.168.1.0/255.255.255.0
```

6. Squid tiene que saber que las máquinas en local\_net ACL tienen acceso al proxy-caché, lo que puede llevar a cabo añadiendo la siguiente línea en squid.conf justo antes de la línea http\_access deny all:

```
http_access allow local_net
```

7. Tiene que decirle a Squid el nombre de usuario del usuario que gestiona el caché. Si quiere utilizar webmaster@yourdomain.com como el usuario que gestiona el caché, tiene que definir la línea siguiente en squid.conf.  
cache\_mgr webmaster

8. Para decirle a Squid qué usuario y qué grupo tiene que ejecutarse, añada las líneas siguientes en squid.conf:

```
cache_effective_user nobody  
cache_effective_group nogroup
```

Aquí le decimos a Squid que se ejecute como el usuario nobody y que utilice permisos para el grupo llamado nogroup.

9. Guarde el archivo squid.conf y ejecute el comando siguiente para crear los directorios caché:

```
/usr/local/squid/squid -z
```

## Iniciar su Squid

Una vez que ha configurado Squid, puede ejecutar el comando /usr/local/squid/bin/squid & para iniciar Squid por primera vez. Puede comprobar que está funcionando de varias formas:

- Squid aparece en una lista ps -x.
- Al ejecutar client www.nitec.com realiza un vuelco de memoria del texto de la página Web en su terminal.
- Los archivos cache.log y store.log del directorio /usr/local/squid/logs muestran que Squid está funcionando.
- Ejecutar squid -k check && echo "Squid is running" le dice que Squid está activo.

Ahora, para realizar la verdadera prueba: si configura el navegador Web en una máquina cliente para utilizar el proxy Squid, debería ver resultados. En

Netscape Navigator, seleccione Editar > Preferencias y entonces seleccione Proxy dentro de la categoría Avanzadas. Seleccionando Configuración manual del servidor proxy y haciendo clic en Ver, puede especificar la dirección IP del servidor Squid como el servidor proxy http, FTP y Gopher. El puerto por defecto para el proxy es 3128, por lo tanto, a no ser que tenga que cambiarlo en el archivo squid.conf, sitúe ese número en el campo del puerto.

**NOTA:** Si utiliza Microsoft Internet Explorer, puede asignar el servidor Squid como su proxy http, FTP y Gopher eligiendo Herramientas>Opciones de Internet>Conexiones>Configuración de LAN. Haga clic en la opción Utilizar un servidor proxy para su LAN, que activará el botón Opciones avanzadas. Haga clic en el botón Opciones Avanzadas e introduzca el servidor Squid y el número de puerto en las cajas de entrada adecuadas para HTTP, FTP y Gopher. Haga clic en OK unas cuantas veces para cerrar todas las cajas de diálogo.

Debería ser capaz de navegar por cualquier sitio Web como si no tuviese un proxy. Puede comprobar que Squid está funcionando adecuadamente comprobando el archivo de registro /usr/local/squid/logs/access.log desde el servidor proxy y asegurándose que el sitio Web que está viendo está ahí.

## **Personalizar Squid para satisfacer sus necesidades**

Ahora que tiene Squid a punto y ejecutándose puede personalizarlo para que se ajuste a sus necesidades. En último término, una herramienta como Squid debería ser totalmente transparente a los usuarios. Esta "invisibilidad" elimina la sensación de complejidad en los usuarios con respecto a la administración, y les permite navegar por la Web como si no estuviesen en un servidor Web proxy. Aunque no voy a dar detalles de cómo hacerlo, puede remitirse a Squid Frequently Asked Questions en <http://squid.nlanr.net/Squid/FAQ/FAQ.html>. La sección 17 de este sitio detalla la utilización de Squid como un proxy transparente.

**NOTA:** Esta sección muestra los conceptos básicos de la utilización de Squid como un proxy Web. Squid tiene muchas características más de las que se discuten aquí. Si está interesado en este tema, visite la página Web en <http://squid.nlanr.net>.

## **Determinar reglas para Squid**

Por defecto, Squid no restringe el acceso a sus usuarios a ningún sitio. Puede definir reglas en su archivo squid.conf para formar listas de control de acceso y permitir o rechazar a visitantes de acuerdo con esas listas; por ejemplo:

```
acl BadWords url_regex foo bar
```

Añadiendo la línea anterior, habrá definido una regla ACL llamada BadWords que corresponde a cualquier URL que contenga las palabras foo o bar.

**NOTA:** Esto se aplica a <http://foo.deepwell.com/pictures> y a <http://www.thekennedycompound.com/ourbar.jpg> porque ambas contienen palabras que son miembros de BadWords.

Añadiendo la siguiente línea:

```
http_access deny BadWords
```

a squid.conf, bloquea el acceso a cualquier URL que se ajuste a esta regla.

**ADVERTENCIA:** Prácticamente todos los administradores que utilizan ACL basadas en palabras tienen un motivo para no examinar todos los modos en los que se puede utilizar una palabra. Puede comprobar que si prohíbe el acceso a sitios que contengan la palabra "sex," también está prohibiendo el acceso a [www.buildersexchange.com](http://www.buildersexchange.com) (exchange significa intercambio) y a otros sitios que no se encuentran dentro de esa categoría.

## Cambiar las opciones de la memoria caché en Squid

Puede controlar la cantidad de páginas Web que puede mantener Squid en la memoria caché.

### Utilizar Redirector en lugar de reglas

Si está gestionando una lista muy grande de sitios de "lista negra" en el archivo squid.conf, debería pensar en utilizar un tipo de programa llamado redirector. Las listas grandes de reglas ACL pueden hacer más lento un proxy Squid. Utilizando un redirector para realizar el mismo trabajo, puede mejorar la eficacia de Squid o permitir o rechazar determinadas URL basadas en reglas de filtro. Puede obtener más información sobre Squirm, un redirector con todas las características, que trabaja con Squid, en [www.senet.com.au/squirm](http://www.senet.com.au/squirm).

El archivo cachemgr.cgi se encuentra en la configuración de Squid. Es un programa CGI que le permite ver estadísticas sobre su proxy, además de apagar e iniciar Squid. Solo necesita unos pocos minutos de su tiempo para instalarlo, y le da detalles explícitos sobre cómo está funcionando su proxy. Si quiere ajustar el caché Web, esta herramienta le ayudará.

Por ejemplo, añadiendo la línea:

```
cache_mem 16 MB
```

puede permitir que Squid utilice 16MB de memoria, para mantener páginas Web en la memoria. Por el método de ensayo y error, podría descubrir si necesita una cantidad distinta.

**ADVERTENCIA:** El `cache_mem` no es la cantidad de memoria que consume Squid; sólo determina la máxima cantidad de memoria que utiliza Squid para mantener páginas Web, imágenes, y cosas de este estilo. La documentación de Squid dice que puede esperar que Squid consuma tres veces más de esta cantidad.

## Escribir registros Squid en formato de registro Apache

Apache escribe registros en el formato habitual de registro, Common Log Format (CLF). Si tiene una herramienta de análisis de registro de Apache, sería perfecto analizar los registros Squid utilizando la misma herramienta. Puede hacerlo, realizando los archivos de registros Squid en el formato CLF.

Utilizando la línea:

```
emulate_httpd_log on
```

los archivos de `/var/log/squid` de forma parecida a los archivos de registro del servidor Web. Este arreglo le permite utilizar programas de estadística Web como Analog o Webtrends para analizar sus registros y examinar los sitios que están viendo sus usuarios.

## Evitar el caching negativo

Si escribe una URL y descubre que la página no existe, hay posibilidades de que la página no vaya a existir en el futuro. Asignando un número determinado de minutos a `negative_ttl`, como se muestra en el siguiente ejemplo, puede controlar cuánto tiempo recuerda Squid que no encontró una página. A esto se le llama caching negativo.

```
negative_ttl 2 minutes
```

El caching negativo no es siempre un buen método. El valor por defecto es de cinco minutos, pero sugiero rebajarlo a dos minutos o a un minuto, o bien desactivarlo totalmente. Esto es para que su proxy sea tan transparente como sea posible. Si un usuario está buscando una página que sabe que no existe, no es necesario que exista un pequeño tiempo muerto entre que la URL aparece y la habilidad del usuario para acceder a ella.

# Utilizar mod\_backhand para una estancia de servidores Web

Si tiene un grupo de servidores (es decir, una estancia de servidores Web) y le gustaría redirigir solicitudes entre los servidores utilizando un módulo nativo de Apache, podría utilizar mod\_backhand. Por ejemplo, si tiene una estancia de servidores Web en tres servidores Apache configurados de forma parecida y le gustaría distribuir alta carga de CGI o solicitudes mod\_perl al que no esté ocupado en ese momento, puede utilizar este módulo. Este módulo utiliza información de estado de recursos de todos los servidores del grupo, y redirige las solicitudes al servidor que esté más preparado para servir una solicitud determinada. A continuación tiene el modo en el que puede bajar, compilar, instalar y configurar este módulo con Apache.

1. Baje la fuente del módulo de [http://www.backhand.org/mod\\_backhand](http://www.backhand.org/mod_backhand).
2. Como ruta, extraiga la distribución fuente en un directorio. Ejecute el comando `./precompile path_to_apache_source` desde el subdirectorio mod\_backhand nuevo. No olvide cambiar `path_to_apache_source` con la ruta actual de la distribución fuente de Apache.
3. Configure la fuente Apache utilizando la opción `--enable-backhand` o `--enable-modules=backhand` y todas sus opciones habituales con el script `configure` desde el directorio de la distribución fuente de Apache.
4. Ejecute `make && make install` como siempre.
5. Añada el siguiente segmento de configuración a su `httpd.conf`:

```
<IfModule mod_backhand.c>
    UnixSocketDir /var/backhand/backhand
    MulticastStats 192.168.1.254:4445
    AcceptStats 192.168.1.0/24
</IfModule>
```

La configuración anterior supone que su dirección IP de amplia difusión es 192.168.1.254 y que tiene una clase de red C, 192.168.1.0/24, que aloja todos sus servidores Web. Asegúrese de cambiar estas direcciones IP para su red. `UnixSocketDir` debe ser accesible únicamente al usuario Apache.

**NOTA:** El módulo mod\_backhand utiliza la dirección Ethernet de amplia difusión para comunicar el estado de los recursos desde cada servidor. El ejemplo anterior utiliza la dirección Ethernet de amplia difusión; puede utilizar direcciones de amplia difusión.

Ahora tiene que decidir en qué directorio quiere equilibrar la carga (es decir, redirigir) entre todos los servidores Web. Normalmente, este es el directorio CGI. Por ejemplo, la siguiente configuración muestra que /www/mysite/cgi-bin tiene que estar bajo el control de mod\_backhand:

```
<Directory "/www/mysite/cgi-bin">
    Backhand byAge
    Backhand byRandom
    Backhand byLogWindow
    Backhand byLoad
</Directory>
```

6. Reinicie el servidor Web Apache utilizando el comando /usr/local/apache/bin/apachectl restart.
7. Repita todos los pasos anteriores para cada servidor Web.

El módulo mod\_backhand crea un proceso demonio que facilita la concurrencia y transmisión de las estadísticas de recursos dentro del grupo de servidores Web. Para garantizar que este demonio da lugar a una partición justa de los recursos del sistema, ejecútelo con alta prioridad. Por ejemplo puede utilizar la utilidad renice que se encuentra en la mayoría de los sistemas Unix y de tipo Unix para asignar esta prioridad en -20.

**NOTA:** Hay otros aspectos específicos de mod\_backhand que tiene que considerar cuando utiliza este módulo. Visite [www.backhand.org/mod\\_backhand](http://www.backhand.org/mod_backhand) para obtener más información.

## Poner a punto aplicaciones Web

Las aplicaciones Web suelen ser la causa principal de los problemas de rendimiento. Las aplicaciones mal configuradas o mal escritas pueden hacer que se cuelguen los servidores. Es muy importante garantizar que sus aplicaciones Web no están causando problemas. Esta sección discute varios trucos que puede utilizar para minimizar los problemas relacionados con las aplicaciones Web. Sin embargo, como las aplicaciones Web se pueden escribir en varios lenguajes utilizando muchos tipos de arquitecturas, es imposible cubrir todos los tipos de aplicaciones Web. Por lo que vamos a limitarnos a los aspectos de las aplicaciones Web basadas en Perl.

### Apurar los scripts mod\_perl

Los scripts mod\_perl pueden aumentar el rendimiento de su aplicación Web porque se cargan una vez y se pueden ejecutar cada vez sin tener que cargar-

las de nuevo. Los siguientes trucos podrían hacer sus scripts mod\_perl aún más rápidos o dar lugar a un mejor rendimiento.

## Precargar sus módulos mod\_perl

Si utiliza módulos mod\_perl para sus sitios Web, plantéese la posibilidad de precargar los módulos utilizando la directiva PerlRequire en httpd.conf. Simplemente, tiene que crear un script Perl que cargue sus módulos habituales. Por ejemplo, a continuación tiene un sencillo script de Perl llamado startup.pl que carga unos cuantos módulos.

```
#!/usr/bin/perl
use CGI ();
use Apache::DBI ();
use Digest::MD5 ();

1;
```

Los paréntesis vacíos desactivan la importación por defecto de las variables del módulo hasta que se necesiten. Esto ahorra memoria porque no debería necesitar todas las características de todos los módulos que precarga. Este script puede precargar en el arranque del servidor utilizando la directiva: PerlRequire /path/to/startup.pl.

Cuando precarga módulos utilizando un script como startup.pl, los procesos hijo pueden al menos compartir un montón de páginas de código utilizado por estos módulos, lo que ahorra RAM, mejorando el estado del sistema.

## Caching conexiones de bases de datos

Si utiliza DBI Perl para acceder a bases de datos relacionales desde sus scripts mod\_perl, puede aumentar el rendimiento de su conexión a la base de datos de manera significativa cambiando una sola línea de código. Si en sus scripts mod\_perl, está utilizando el use DBI; realice una llamada para utilizar directamente el módulo DBI, entonces cambie esto a use Apache::DBI;, que cacheará las conexiones a la base de datos para su aplicación e incrementará el rendimiento de forma significativa. Hay muchas otras técnicas de programación que también tiene que considerar:

- Si conecta con la misma base de datos para cada solicitud, entonces considere abrir la conexión a la base de datos fuera del manejador de solicitudes. Por ejemplo:

```
sub handle {
    my $r = shift;

    my $q = new CGI;
    my $id = $q->param('id');
```

```

# Conecta con la base de datos

my $dbh;

eval {

    $dbh = DBI->connect($dataSource,
                          $dbUser,
                          $dbPassword, { AutoCommit -> 1 }) || 
        die;
};

if ($@) {
    # muere "No puede conectar con la base de datos
    # $DBI::errstr";
    # conexión fallida
    print STDERR "Can not connect to $dataSource \n";
    print STDERR "DB2 Server Error: $DBI::errstr \n";
}

my $statement = "SELECT myfield from mytable where ID =
$id";

my $sth = $dbh->prepare($statement);

my $rv = $sth->execute;

if (! $rv ) {
    # No encuentra registro. Tiene que hacer algo
}
}

while ( my @fields = $sth->fetchrow_array){

    print STDOUT "ID $id shows: ", join(',', @fields),
    "<br>";

}

$sth->finish;
}

```

Observe que la conexión a la base de datos se realiza dentro del manejador, que puede realizar fuera del manejador porque las mismas credenciales de bases de datos y usuarios se utilizan para la conexión.

- Si es posible, prepare las sentencias SQL una vez y reutilice las sentencias preparadas para ahorrar tiempo. El ejemplo anterior se puede escribir de forma más eficaz:

```

my $APP_RUN_COUNT = 0;
my ($dbh, $sth);

sub init {
    eval {
        $dbh = DBI->connect($dataSource,
            $dbUser,
            $dbPassword, { AutoCommit => 1 }) || die;
    };
    if ($@) {
        # muere "No puede conectar con la base de datos
        # $DBI::errstr";
        # conexión fallida
        print STDERR "Can not connect to $dataSource \n";
        print STDERR "DB2 Server Error: $DBI::errstr \n";
    }
}

my prepare_statement {
    my $dbh = shift;

    my $statement = "SELECT myfield from mytable where ID =
?";
    $sth = $dbh->prepare($statement);
}

sub handle {
    my $r = shift;

    if ($APP_RUN_COUNT++ == 0) {

        init();
        prepare_statement();

    }

    my $q = new CGI;
    my $id = $q->param('id');

    my $rv = $sth->execute($id);

    if (! $rv ) {

```

```

        # No se encuentra ningún registro.
    }

while ( my @fields = $sth->fetchrow_array) {
    print STDOUT "ID $id shows: ", join(',', @fields),
    "<br>";
}

$sth->finish;
}

```

Aquí el método del manejador llama a las rutinas `init()` y `prepare_statement()` para crear bases de datos globales y manejadores de sentencias para el ciclo de vida completo de los procesos del servidor. Esto hace que el script sea mucho más eficaz que en la versión anterior.

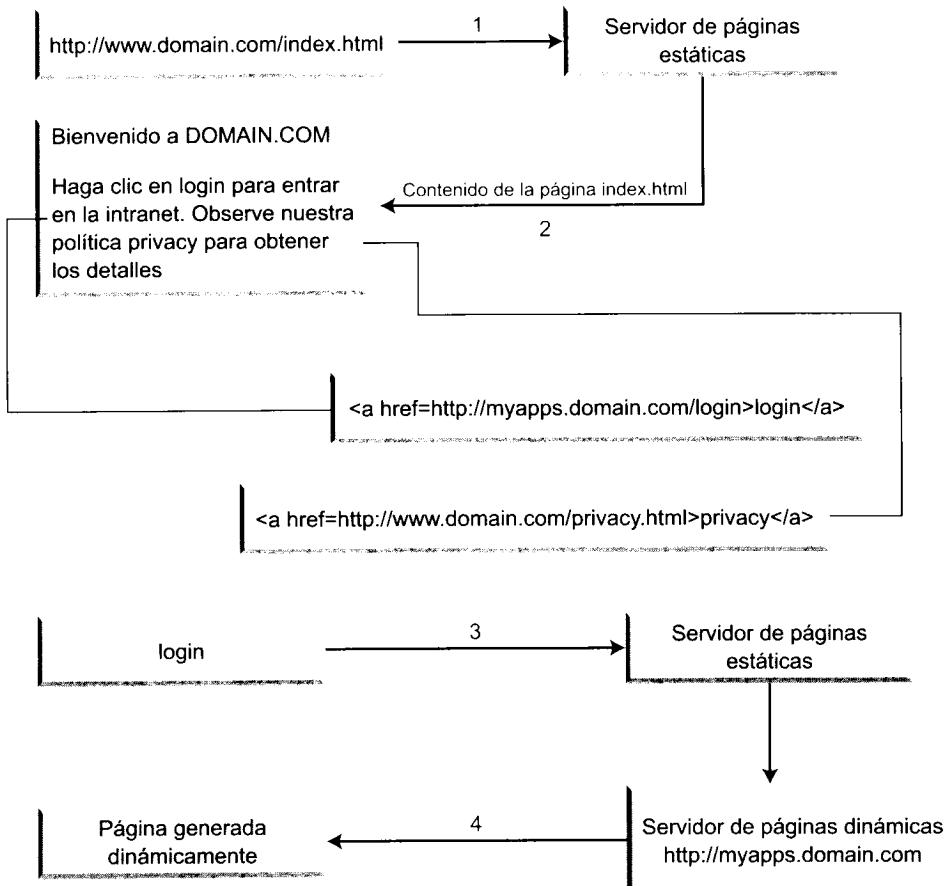
## **Ejecutar aplicaciones mod\_perl en un conjunto parcial de hijos Apache**

Cuando utilizamos muchos scripts `mod_perl`, observamos que los procesos hijo del servidor Apache se hacen más grandes. Puede ser testigo de este fenómeno utilizando el comando `top`. Mientras tenga suficiente RAM puede estar tranquilo. Sin embargo, nadie tiene nunca suficiente RAM, y por lo tanto es una buena idea no confiar en tener mucha memoria como solución y pensar en cómo puede dirigir este problema de forma más eficaz.

Si descubre que los procesos hijo Apache se hacen más y más grandes porque muchos scripts `mod_perl` están cargados en ellos, considere la posibilidad de tener un servidor de script dedicado que únicamente sirva contenido dinámico. La figura 22.6 muestra cómo funciona.

Cuando un usuario solicita la página home de un sitio llamado `www.domain.com`, el servidor Apache responsable de las páginas estáticas devuelve la página `index.html` al cliente. La página contiene enlaces embebidos para contenido estático y para contenido dinámico. La figura muestra dos enlaces de este tipo: `login` y `privacy`. Cuando un usuario final hace clic en el enlace `login` está solicitando `http://myapps.domain.com/login`, que es un servidor Apache distinto al servidor `www.domain.com`. De hecho, deberían ser dos sistemas Linux distintos en un mundo ideal. Sin embargo, no todo el mundo puede afrontar la escisión del contenido estático y dinámico, y esta solución no es apropiada para todo el mundo. Si tiene que mantener `mod_perl` y el contenido estático en el mismo sistema Linux que ejecuta Apache, puede seguir garantizando que esos procesos hijo Apache grandes no están sirviendo páginas estáticas. A continuación tiene la solución que más me gusta:

1. Compile e instale el módulo mod\_proxy para su servidor Web Apache.
2. Copie su archivo httpd.conf en httpd-8080.conf y modifique la directiva Port para que sea Port 8080 en lugar de Port 80. Elimine todas las configuraciones específicas de mod\_perl desde httpd.conf de modo que todas sus configuraciones mod\_perl se encuentren en el archivo httpd-8080.conf.



**Figura 22.6.** Separar contenido estático del dinámico (script generado por mod\_perl)

3. Modifique el archivo httpd.conf para tener las siguientes directivas proxy:

```
ProxyPass /myapps http://127.0.0.1:8080/myapps
```

Puede cambiar myapps con lo que quiera. Si realiza este cambio, tiene que cambiarlo también en cualquier otra localización en la que hayamos mencionado esta discusión. Aquí, el servidor Apache que está sirviendo

páginas estáticas, va a ser instruido para que todas las solicitudes a la URL /myapps se sirvan mediante el módulo proxy, que debería obtener la respuesta del servidor Apache en el mismo sistema Linux (127.0.0.1 es el host local) pero en el puerto 8080.

4. Añada la siguiente configuración en httpd-8080.conf para crear un script location en mod\_perl.

```
<Location /myapps>
    SetHandler perl-script
    PerlHandler MyApp1
</Location>
```

No olvide cambiar MyApp1 con el nombre de su propio script.

5. Si tiene un KeepAlive On en httpd-8080.conf, cámbielo a Off. Esto garantiza que Apache no mantiene la conexión abierta durante el número de segundos especificados en KeepAliveTimeout, con la esperanza de servir a nuevos clientes desde la misma conexión TCP.
6. Inicie (o reinicie) el servidor Apache (que se escuche en el puerto 80) de la forma habitual, utilizando el comando apachectl. Sin embargo, tiene que iniciar el Apache en el puerto 8080 utilizando el comando /usr/local/apache/bin/httpd -f /usr/local/apache/conf/httpd-8080.conf. Esto supone que tiene instalado el directorio /usr/local/apache; si eso no es verdad, cambie la ruta.

Ahora tiene dos padres Apache demonios (es decir, ejecutándose como raíz) ejecutando dos conjuntos de procesos hijo en los que uno sirve páginas estáticas y utiliza el módulo proxy para buscar las páginas dinámicas del script de mod\_perl utilizando la directiva ProxyPass. Esto le permite servir páginas estáticas utilizando un conjunto de servidores hijo que no están ejecutando código Perl. Por otro lado, el servidor en el puerto 8080 únicamente sirve solicitudes dinámicas; por eso, tiene una configuración de alto rendimiento.

## Utilizar FastCGI en lugar de mod\_perl

Los scripts que se ejecutan bajo mod\_perl son rápidos porque se cargan dentro del espacio de código de cada servidor hijo. A diferencia de su homólogo, un script mod\_perl puede mantener una conexión permanente con un servidor externo de bases de datos. Esto significa que la generación de contenido dinámico dirigida por bases de datos se vuelve rápida con los scripts mod\_perl.

Sin embargo, se introduce un problema nuevo si ejecuta un servidor muy grande. Cuando ejecutamos 50 o más procesos en un servidor Apache para servir muchas solicitudes simultáneas, es posible para Apache abrir a la larga todas esas conexiones y mantener cada conexión persistente para la duración de cada proceso hijo. Imagine que quiere ejecutar un sistema servidor Web en el que se

ejecutan 50 procesos hijo Apache de modo que puede servir aproximadamente 50 solicitudes por segundo y que tiene un script basado en `mod_perl` que abre una conexión a una base de datos en la fase inicial. Según llegan las solicitudes al script de su base de datos, Apache sirve esas solicitudes utilizando sus procesos hijo y abriendo 50 conexiones a la base de datos. Como muchos servidores de bases de datos tienen recursos caros para las conexiones, esto puede ser un aspecto importante en el lado de la base de datos.

Por ejemplo, cuando realizamos este tipo de conexiones con un IBM Universal Database Server (UDB) Enterprise Edition ejecutándose en un sistema Linux, cada hijo Apache tiene un proceso homólogo relacionado con la conexión en el servidor de la base de datos. Si ese entorno utiliza hardware de equilibrio de carga para equilibrar las solicitudes entrantes entre un conjunto de servidores Apache compatibles con `mod_perl`, se trata prácticamente de un escenario en el que cada sistema Web-servidor, que está ejecutando 50 procesos hijo de Apache, tiene todos los hijos Apache abiertos y conectados al servidor de la base de datos. Por ejemplo, si ese ambiente consiste en 10 servidores Web bajo el hardware de equilibrio de carga, entonces el número posible de conexiones con el servidor de la base de datos es  $10 \times 50$ , o 500 conexiones, lo que podría dar lugar a una carga amplia de recursos en el servidor de la base de datos.

Una posible solución para esta situación es encontrar un modo de tener conexiones desocupadas durante los tiempos muertos, hacer que el código del script `mod_perl` detecte estas conexiones, y hacer que el código reinicie la conexión. Otra solución es crear un demonio proxy persistente en la base de datos, de modo que cada servidor Web lo utilice para extraer datos de la base de datos.

FastCGI o los Java Servlets tienen más de una solución nativa para este tipo de problemas y deberían considerarse para las aplicaciones manejadas por bases de datos. La siguiente sección discute otra tecnología Web que aumenta el rendimiento llamada FastCGI.

Al igual que los scripts `mod_perl`, las aplicaciones FastCGI se ejecutan todo el tiempo (tras la carga inicial) y por lo tanto proporcionan un aumento importante de rendimiento con respecto a los scripts CGI. La tabla 22.2 explica las diferencias entre una aplicación FastCGI y un script `mod_perl`.

A lo largo de este libro, aprenderá más sobre FastCGI:

**Tabla 22.2.** Diferencias entre una aplicación FastCGI y los scripts `mod_perl`

Aspecto	Aplicaciones FastCGI	Scripts <code>mod_perl</code>
Dependiente de la plataforma Apache.	No. Las aplicaciones FastCGI pueden ejecutarse en un servidor Web que no sea Apache como IIS, el servidor Web Netscape, y similares.	Sí. Sólo Apache soporta el módulo <code>mod_perl</code> .

Aspecto	Aplicaciones FastCGI	Scripts mod_perl
Solución únicamente para Perl.	No. Las aplicaciones FastCGI pueden desarrollarse en muchos lenguajes incluyendo C, C++ y Perl.	Sí.
Se ejecuta como un proceso externo.	Sí.	No.
Se puede ejecutar en una máquina remota.	Sí.	No.
Se ejecutan varias instancias de aplicación/script.	Normalmente, se ejecuta una sola aplicación FastCGI en respuesta a varias solicitudes en cola. Sin embargo, si la carga es alta, se ejecutan varias instancias de la misma aplicación.	El número de instancias de scripts mod_perl que se ejecutan es igual al número de procesos hijo del servidor Apache.
Gran soporte disponible.	Sí. Sin embargo, a veces tengo la impresión de que el desarrollo de FastCGI es cada vez más lento pero no puedo comprobarlo ni realizar un backup.	Sí. Hay una gran de sitios mod_perl en Internet y hay soporte disponible mediante Usenet o Web.
Conectividad a bases de datos.	Como todas las solicitudes se envían a una sola aplicación FastCGI, si tiene que mantener una sola conexión a la base de datos con el servidor de la base de datos del lado del servidor. Sin embargo, esto puede cambiar cuando la gestión del proceso FastCGI produce más instancias de aplicaciones FastCGI debido a la gran carga. Pero tranquícese, el número de instancias FastCGI de una aplicación es menor que el número de procesos hijo.	Como cada proceso hijo Apache ejecuta el script mod_perl, cada hijo puede tener potencialmente una conexión con la base de datos del lado del servidor. Esto significa que podría tener cientos de conexiones a la base de datos desde, incluso, un solo sistema servidor de bases de datos.



# **23** Crear una red de alta disponibilidad

---

## **En este capítulo**

1. Entendemos las características de una red Web.
2. Incrementamos la seguridad DNS.
3. Realizamos un equilibrio de carga en una red Web.
4. Gestionamos el almacenaje Web.
5. Creamos una red back-end para mantenimiento.
6. Fortificamos su red Web.

En este capítulo, aprenderá algunas consideraciones de diseño para construir una red Web. Una red Web es una red de servidores Web que crea un servicio Web. Por ejemplo, Yahoo! Utiliza un gran número de servidores Web, servidores de aplicación y servidores de bases de datos para crear una gran cantidad de servicios Yahoo!

Si ha decidido integrar la Web en su negocio, debe tener en cuenta una solución Web que pueda crecer más allá de un solo servidor Web o de un espacio de disco Web compartido en una ISP. Una red Web es una solución. En este capítu-

lo, aprenderá a utilizar red y conceptos de gestión de red con respecto a las consideraciones de diseño para construir una red Web adecuada. Aunque el capítulo se centra en soluciones basadas en Apache en Linux, la mayoría de las consideraciones se pueden aplicar a otras plataformas.

## Características de una red de alto nivel

Una red de alto nivel sirve de miles a millones de páginas al día. Para servir un gran número de solicitudes Web debe tener las siguientes características:

- **Servidores DNS seguros.** Si sus servidores Domain Name Service (DNS) se cuelgan, nadie podrá acceder fácilmente a sus sitios Web. Por eso, la seguridad de DNS es una consideración importante en el diseño de una red Web.
- **Acceso Web con equilibrio de carga.** Los usuarios conectan con uno o más servidores Web, que son seleccionados automáticamente en función de la carga del sistema y la disponibilidad.
- **Una arquitectura de almacenaje gestionable.** Una red Web tiene que tener una gran cantidad de contenido, que ha de ser seguro, disponible y ser fácilmente gestionable. Tener varios servidores y gestionar el disco duro de cada servidor puede ser una pesadilla sin la planificación y una sólida arquitectura de almacenaje.
- **Redes back-end eficaces.** Los sitios Web grandes ejecutan muchas aplicaciones y realizan millones de consultas a bases de datos, así como muchas otras tareas del lado del servidor, para dar lugar a alta calidad y a contenido personalizado para sus clientes. Por tanto es necesaria una red sólida del lado del servidor. Una red back-end es una red a la que no pueden acceder los visitantes Web pero que sirve de espina dorsal para mantener cada servidor actualizado y sincronizado con el contenido. Las redes back-end también le permiten al administrador realizar varias tareas de administración del sistema como son el backup, la actualización del software, etc.
- **Alto grado de seguridad.** Los diseñadores deben poner atención en los aspectos de seguridad cuando diseñan redes Web porque los piratas informáticos atacan a menudo las redes Web que tienen brechas de entrada y utilizan estas redes Web como plataformas para atacar otros sitios. Esto puede dar lugar a un problema legal muy serio.

## Aumentar la seguridad DNS

Cuando se cuelga un sitio DNS, resulta inalcanzable para la mayoría de los usuarios. Si conoce la dirección IP de un sitio Web probablemente pueda acceder

a él, pero la mayor parte de la gente no sabrá cómo determinar cuál es la dirección IP de un sitio, por lo que no visitarán su sitio Web.

Por este motivo son necesarios dos servidores DNS para registrar nuevos dominios. Sin embargo, debería considerar este requisito como un mínimo absoluto y debería utilizar más de dos servidores para aumentar la redundancia y la seguridad de su servicio DNS. A continuación le presento la estrategia DNS que suelo recomendar:

- Utilice al menos dos servidores DNS para su sitio o sitios Web.
- Utilice un servidor DNS secundario fuera del sitio. Esto significa que si su servidor DNS local se apaga, el servidor DNS secundario seguirá siendo funcional. Por ejemplo, si su servidor DNS principal está situado en una red que está fuera de servicio temporalmente, el servidor secundario responderá y dirigirá el tráfico a los recursos adecuados. Si no quiere tener un servidor DNS secundario y mantenerlo usted mismo, puede utilizar un servicio como `secondary.com`.
- Utilice al menos un servidor DNS dedicado, si es posible, porque los sistemas que ejecutan varios servicios tienen más probabilidades de colgarse con frecuencia que los que ejecutan un solo servicio.
- Ejecute un caché local DNS si alguna de sus aplicaciones Web necesitan traducir direcciones IP DNS en nombres de host. Por ejemplo, hay sistemas de anuncios comerciales que utilizan direcciones IP que corresponden a zonas demográficas realizando búsquedas DNS asincrónicas. Este tipo de aplicaciones pueden utilizar el caching DNS.
- Utilice monitorización del software sobre la base de una rutina para garantizar que el dato DNS es correcto y se encuentra disponible.

## **Equilibrio de carga en su red**

El motivo de tener varios servidores Web en una red Web es equilibrar la carga entre ellos para garantizar un alto grado de rendimiento y estabilidad. Hay dos formas principales de equilibrar la carga entre servidores: Round-Robin DNS o retorno al punto de origen o barrido cíclico, y equilibrador de carga basado en hardware. Ambos métodos se discuten en las secciones siguientes.

### **Distribuir solicitudes HTTP con Round-Robin DNS**

La solución Round-Robin DNS se recomienda únicamente si no se puede utilizar una solución con un equilibrador de carga basado en hardware. Round-Robin

DNS es un mecanismo para buscar un nombre de host cíclicamente en una lista de direcciones IP de servidores Web.

Suponga que tiene dos servidores Web, `www1.yourdomain.com` (192.168.1.10) y `www2.yourdomain.com` (192.168.1.20), y que quiere equilibrar la carga para `www.yourdomain.com` en estos dos servidores utilizando el truco Round-Robin DNS. Entonces siga los pasos siguientes:

1. Añada las siguientes líneas en el archivo `yourdomain.com`:

```
www1 IN A 192.168.1.10  
www2 IN A 192.168.1.20
```

```
www IN CNAME www1  
www IN CNAME www2
```

2. Reinicie su servidor DNS y realice un ping al host `www.yourdomain.com`. Verá la dirección 192.168.1.10 en la salida del ping.
3. Pare el ping y vuelva a realizar un ping para el mismo host, entonces verá la segunda dirección IP, porque la configuración anterior le dice al nombre del servidor que pase cíclicamente por los registros CNAME buscando `www`. En otras palabras, el host `www.yourdomain.com` es tanto `www1.yourdomain.com` como `www2.yourdomain.com`.

Cuando alguien introduce `www.yourdomain.com`, el nombre del servidor reparte la primera dirección una vez, entonces reparte la segunda dirección para la segunda solicitud, y mantiene el ciclo entre estas direcciones.

**NOTA:** Una desventaja del truco Round-Robin es que el servidor DNS no puede saber qué sistema tiene mucha carga y cuál no, se trata de ciclos ciegos. Si uno de los servidores se estropea o deja de estar disponible por alguna razón, el truco Round-Robin DNS sigue devolviendo la IP de ese servidor. Esto significa que algunas personas serán capaces de acceder a los sitios y otras no.

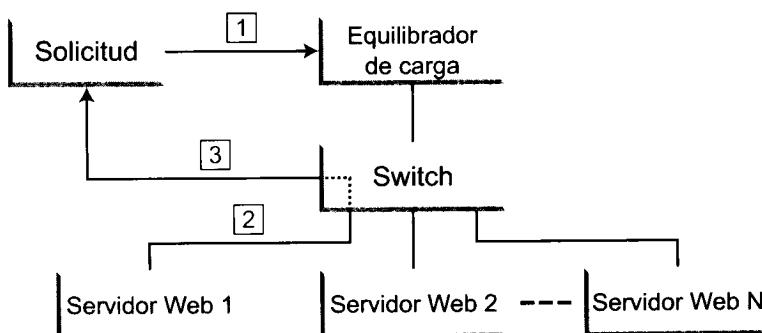
## Distribuir solicitudes HTTP con equilibradores de carga basados en hardware

Los equilibradores de carga basados en hardware son muy habituales. Una solución de equilibrio de carga con un hardware es mucho más funcional que la solución Round-Robin DNS discutida en la sección anterior. Un dispositivo de hardware para equilibrio de carga puede implementar varias formas de monitorizar cada una de las cargas, el rendimiento y la disponibilidad de la red de servidores Web, utilizando el tiempo de respuesta, el número de solicitudes enviadas y reali-

zando solicitudes HTTP de prueba auto generadas. Por lo tanto, estos dispositivos ofrecen un mayor control para su esquema de equilibrio de carga.

Algunos de los dispositivos de equilibrio de carga también están disponibles para crear grupos de servidores en los que algunos servidores tengan más prioridad que otros. Por ejemplo, si tiene un sistema Pentium 4 1.3 GHz con 2GB de RAM y un sistema Pentium III 550 MHz con 512GB de RAM, puede darle mayor prioridad al sistema Pentium 4, ya que es más poderoso y es más probable que sea capaz de servir muchas más solicitudes que el sistema Pentium III. Local Director (CISCO) y Web Director (Radware) son soluciones hardware de equilibrio de carga que funcionan muy bien.

La figura 23.1 muestra una solución sencilla de equilibrio de carga en la que cada solicitud del cliente llega al hardware de equilibrio de carga.



1: La solicitud del cliente llega al equilibrador de carga

2: El equilibrador de carga selecciona un servidor Web disponible

3: El servidor Web seleccionado responde a la solicitud del cliente

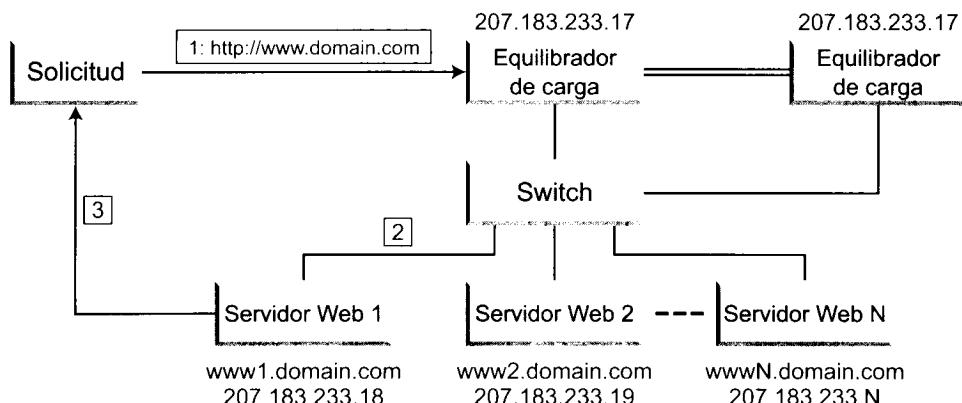
**Figura 23.1.** Una solución de equilibrio de carga sencilla

El equilibrador de carga decide qué servidor Web utilizar para servir la solicitud y pasarla al servidor más adecuado, que responderá a la solicitud de la forma habitual. El criterio de selección para el servidor Web puede depender de la prioridad, la disponibilidad y la seguridad.

Se puede observar que esta solución tiene un solo punto de entrada en la red Web. Por ejemplo, las solicitudes para `http://www.domain.com` deben llegar al equilibrador de carga, que internamente la traduce a una solicitud para que la sirva el servidor Web de la red. La figura 23.2 muestra un ejemplo de una red Web con la carga equilibrada.

Una solicitud para `http://www.domain.com` se envía al equilibrador de carga porque `www.domain.com` traduce la dirección IP del equilibrador de

carga. En otras palabras, debe establecer que los registros DNS apunten a `www.domain.com` en su sistema de equilibrio de carga. El equilibrador de carga decide entonces cuál de los servidores Web `www[1-N].domain.com` responderá a la solicitud.



**Figura 23.2.** Una solución de equilibrio de carga sencilla para `www.domain.com`

Observando la figura 23.2, observará que hay una segunda conexión del equilibrador de carga con el switch de la red interna, así como una conexión directa entre el equilibrador de carga principal (a la izquierda) y el equilibrador de carga secundario (a la derecha). La conexión directa es un cable cruzado de red o una conexión RS232 en serie. Los dispositivos de hardware utilizan esta conexión para mantener el estado. En otras palabras, el equilibrador secundario sigue la pista de cada una de las operaciones realizadas por el equilibrador principal. Si el primer equilibrador de carga deja de estar disponible por un fallo de hardware, el equilibrador de carga secundario inicia respondiendo a la misma dirección IP que el principal y se hace cargo del trabajo. La conexión cruzada garantiza que el equilibrador de carga secundario tenga acceso al estado de la red y que por lo tanto sea capaz de restablecer el estado actual de conexiones entre el sitio y los clientes.

Además tenga en cuenta que si utiliza la herramienta de traducción NAT en su hardware de equilibrio de carga (si está equipado con esta característica), no necesita direcciones IP ruteables para cada uno de sus servidores Web.

## Gestionar almacenamiento Web

Como desarrollador de redes Web, debe tener una estrategia de almacenamiento claro y definida. Una gestión de almacenamiento es crítica para las redes Web. Hay muchas tecnologías a tener en cuenta, y muchos aspectos de rendimiento que dirigir en la creación de una arquitectura sólida de almacenamiento para su red.

Web. Esta sección discute algunas de las principales tecnologías de almacenamiento, así como las técnicas de rendimiento prácticas para potenciar su rendimiento de almacenamiento bajo la plataforma Linux.

## **RAID, SAN o dispositivos de almacenamiento**

Los sistemas redundantes de discos independientes (Redundant Array of Inexpensive or Independent Disks, RAID), las redes de almacenamiento (Storage Area Network ,SAN), o los dispositivos de almacenaje, son las tecnologías de almacenamiento más utilizadas. Todos ellos aumentan la seguridad o la seguridad y el rendimiento. Elegir la solución adecuada para sus necesidades exige un entendimiento profundo de estas tecnologías, que se discuten en las secciones siguientes.

### **RAID de hardware**

Una solución RAID de hardware utiliza unidades de disco SCSI con una tarjeta de control RAID. Independientemente del tipo de RAID (hardware o software) que utilice, tendrá que seleccionar un nivel RAID adecuado para sus necesidades. Los niveles más habituales para RAID son 1 y 5. RAID 1 es una pura redundancia de unidades de disco. Si quiere utilizar redundancia RAID 1 de unidades de discos y quiere tener un espacio total de 100GB, tiene que introducir 200GB de espacio en la unidad de disco. Aunque están disponibles los niveles de 2 a 4 de RAID, RAID 5 suele ser la mejor elección. Si utiliza N dispositivos de tamaño S, el tamaño del array completo será  $(N-1)*S$ . El espacio sobrante se utiliza para paridad de información (redundancia). Se recomienda utilizar el mismo tamaño para no gastar el espacio de la unidad de disco porque se utiliza el disco más pequeño entre N discos para calcular el espacio de disco bajo RAID 5.

### **Las redes de almacenamiento (SAN)**

Las redes de almacenamiento (SAN) son el santo grial de las soluciones de almacenaje. Compañías como EMC, IBM, Compaq y Storage Networks son proveedores de SAN. Normalmente, una solución SAN consiste en dispositivos dedicados de almacenamiento en una red de fibra óptica y el almacenaje se hace disponible para sus sistemas Linux mediante un switch de hardware y una tarjeta de interfaz de canal de fibra. Hablando en términos generales, SAN es para el mundo de la empresa y no está disponible para las organizaciones pequeñas y medianas.

Sin embargo, si sitúa sistemas Linux en un centro de datos bien conocido como los centros proveedores de grandes ISP, como Exodus y Globix, hay posibilidades de que encuentre SAN como un servicio de valor añadido. Esto podría ser una forma de no pagar por un hardware de SAN, simplemente tendrá acceso a él. Hay redes de almacenamiento que proporcionan este tipo de servicios en las principales ISP. También tienen anillos de fibra a lo largo de los Estados Unidos, lo que significa que puede hacer que sus unidades de disco de New York aparezcan en California con un retardo despreciable.

## **Dispositivos de almacenamiento**

Actualmente tenemos hardware especial para cada función, de modo que los dispositivos de almacenamiento (que son sistemas de almacenamiento dedicados) suelen ser una solución habitual para los administradores de red y de sistemas. Hoy, puede comprar dispositivos de almacenamiento dedicados que transmiten en cadena 10, 100 o 1000Mbits Ethernet y proporcionan servicios de almacenaje RAID. Estos dispositivos se suelen gestionar de forma remota a través de la Web. Son adecuados para organizaciones pequeñas y medianas y suelen ser fáciles de gestionar y de configurar.

## **Poner a punto sus discos duros**

Independientemente del tipo de solución de almacenamiento que elija para su red Web, tiene que tratar con discos duros. Puede utilizar discos de canal de fibra, SCSI o IDE para implementar sus soluciones de almacenaje. De hecho, SCSI y IDE son los tipos de disco duro más comunes en el mundo actual de la computación. Los discos SCSI y los controladores SCSI son mucho más caros que los discos IDE porque proporcionan mayor rendimiento y flexibilidad. IDE o la versión mejorada de IDE llamada discos EIDE son los más habituales en la computación personal y en la dirigida por I/O no intensiva. La diferencia entre el mundo SCSI y el mundo IDE es que los controladores de discos SCSI manejan la mayoría del trabajo de transferencia de datos desde y hacia los discos, mientras que la propia CPU controla los discos IDE. Por lo tanto, en un sistema de discos SCSI muy ocupado, no puede añadir tanta carga en la CPU como hacen los discos IDE. Además, los discos SCSI tienen más capacidades de transferencia, mientras que los discos IDE siguen conectados al sistema mediante un bus de 16-bit. Si necesita mayor rendimiento, SCSI es la forma de hacerlo. Compre adaptadores SCSI de marcas registradas y RPM ultrawide de 10K o discos SCSI más veloces, y habrá hecho prácticamente todo lo que puede hacer para aumentar su subsistema de discos.

**TRUCO:** Por supuesto, si tiene presupuesto suficiente, puede utilizar discos de canal de fibra o utilizar una solución SAN. La última opción es la que se utiliza normalmente en empresas con alta demanda de almacenamiento de datos. Puede utilizar también una solución hardware / software RAID, que se ha discutido ya en este capítulo.

Independientemente de su decisión con respecto a utilizar discos SCSI o IDE, debería utilizar varios discos y obtener un rendimiento adecuado. Como mínimo, debería utilizar dos discos, uno para el sistema operativo y el software y el otro para los datos. Para los servidores Web, normalmente recomiendo un mínimo de tres discos. El tercer disco se utiliza para los registros generados por los sitios

Web alojados en la máquina. Tener discos I/O repartidos en varios dispositivos garantiza un tiempo de espera mínimo.

**NOTA:** Si tiene un disco ultrawide SCSI moderno establecido para su sistema Linux, puede obtener un gran rendimiento de sus unidades de discos.

## Obtener hdparam

Para aumentar el rendimiento de su disco EIDE moderno, debe determinar primero el rendimiento actual de su disco antes de realizar cualquier ajuste. Por lo tanto, necesita una herramienta de medida del rendimiento de su sistema de discos. La herramienta hdparam es perfecta para este trabajo; puede bajar la distribución fuente de <http://metalab.unc.edu/pub/Linux/system/hardware/>. Para compilar e instalar la herramienta hdparam, siga estos pasos:

1. Establézcase como raíz.
2. Extraiga la distribución fuente en un directorio adecuado como /usr/local/src. Por ejemplo, y ejecute el comando tar xvzf hdparm-3.9.tar.gz en /usr/local/src para extraer la distribución fuente de la versión 3.9 de hdparam.
3. Cambie al nuevo subdirectorio creado y ejecute el comando make install para compilar e instalar el binario hdparam y la página del manual. Por defecto, el binario se instala en el directorio /usr/local/sbin y se llama hdparam.

**ADVERTENCIA:** Como hdparam le permite cambiar el comportamiento del sistema de discos IDE/EIDE, puede dar lugar a que el sistema se cuelgue debido a un mal uso o a una configuración inadecuada. Le recomiendo realizar una copia de los datos antes de utilizar hdparam. Además, es una buena idea experimentar con hdparam en el modo de un solo usuario. Puede reiniciar el sistema y forzarlo para que utilice el modo de un solo usuario introduciendo linux single en el prompt lilo durante en reinicio.

## Estimar el rendimiento de su unidad de disco

Una vez que ha instalado la herramienta hdparam, está preparado para investigar el rendimiento del sistema de discos. Suponiendo que su disco duro IDE

o EIDE es /dev/hda, ejecute el siguiente comando para ver el estado de la configuración de su disco duro:

```
hdparm /dev/hda
```

Obtendrá una salida parecida a la siguiente:

```
/dev/hda:  
multcount      =  0 (off)  
I/O support    =  0 (default 16-bit)  
unmaskirq     =  0 (off)  
using_dma      =  0 (off)  
keepsettings   =  0 (off)  
nowerr         =  0 (off)  
readonly        =  0 (off)  
readahead       =  8 (on)  
geometry       = 2494/255/63, sectors = 40079088, start = 0
```

Como puede ver, está desactivado prácticamente todo por defecto. Algunos de estos valores por defecto se tienen que cambiar para aumentar el rendimiento del sistema. Antes de continuar el proceso, necesita más información para el disco duro. Ejecute el comando siguiente:

```
hdparm -i /dev/hda
```

Este comando devolverá una información parecida a la siguiente:

```
/dev/hda:  
  
Model=WDC WD205AA, FwRev=05.05B05, SerialNo=WD-WMA0W1516037  
Config={ HardSect NotMFM HdSw>15uSec SpinMotCtl Fixed DTR>5Mbs  
FmtGapReq }  
RawCHS=16383/16/63, TrkSize=57600, SectSize=600, ECCbytes=40  
BuffType=DualPortCache, BuffSize=2048kB, MaxMultSect=16,  
MultSect=16  
CurCHS=16383/16/63, CurSects=16514064, LBA=yes,  
LBAsects=40079088  
IORDY=on/off, tPIO={min:120,w/IORDY:120},  
tDMA={min:120,rec:120}  
PIO modes: pio0 pio1 pio2 pio3 pio4  
DMA modes: mdma0 mdma1 *mdma2 udma0 udma1 udma2 udma3 udma4
```

El comando anterior muestra la información de identificación (si existe) de unidades de discos duros que estaba disponible durante la última vez que inició el sistema. Va a necesitar esta información más tarde. El comando anterior está informando sobre el modelo, la configuración, la geometría de la unidad de disco (cilindros, cabezales, sectores), tamaño de pista, tamaño del sector, tamaño del buffer, modo soportado de DMA, modo PIO, y similares. Utilizando el siguiente comando puede comprobar el sistema de unidades de disco:

```
/usr/local/sbin/hdparm -Tt /dev/hda
```

Verá un resultado parecido a este:

```
/dev/hda:  
Timing buffer-cache reads: 128 MB in 1.01 seconds = 126.73 MB/sec  
Timing buffered disk reads: 64 MB in 17.27 seconds = 3.71 MB/sec
```

Por supuesto, sus números variarán dependiendo de su unidad de disco y del subsistema de control. Este es el estado sin ajustar de su subsistema de unidades de disco. La opción **-T** le dice a **hdparam** que compruebe el subsistema caché (es decir, la memoria, la CPU, y el caché buffer). La opción **-t** le dice al programa **hdparam** que informe sobre el estado de la unidad de disco (**/dev/hda**) leyendo datos que no se encuentren en el caché. Ejecute este comando durante algunos minutos y tome la media de MB/seg que se obtiene para su unidad de disco. Este sería en rasgos generales, el estado de rendimiento de su subsistema de unidades de disco. En este ejemplo podemos leer un rendimiento de 3.71 MB/seg, que es un número muy bajo.

## Mejorar el rendimiento de su unidad de disco

A continuación vamos a intentar mejorar el rendimiento de su unidad de disco. Observando la salida del comando **hdparam -i /dev/hda** (remítase a la sección anterior) y busque el valor de **MaxMultSect**. En este ejemplo es 16. El comando **hdparam /dev/hda** muestra un valor de 0 (off) para **multcount**. Esto significa que el modo multisector (es decir, modo bloque IDE) está desactivado.

El modo multisector es una característica de la mayoría de los discos duros IDE modernos. Le permite al disco transferir varios sectores de disco por cada interrupción I/O. Por defecto, se encuentra desactivado. Sin embargo, la mayoría de los discos modernos pueden realizar 2, 4, 8 o 16 transferencias de sectores por cada interrupción I/O. Por eso, si fija este modo en el máximo valor posible para su disco, que muestra el valor de **MaxMultiSect**, verá un aumento en la capacidad de procesamiento del 5 por 100 al 50 por ciento o más. Además, reducirá la sobrecarga del sistema operativo de un 30 a un 50 100. En este ejemplo, el valor de **MaxMultiSect** es 16, de modo que podemos utilizar la opción **-m** de la herramienta **hdparam** para asignarlo y ver cuánto aumenta el rendimiento. Ejecute el comando siguiente:

```
/usr/local/sbin/hdparm -m16 /dev/hda
```

Ahora ejecute la prueba de rendimiento utilizando el comando **hdparam -tT /dev/hda** para ver el cambio. Para el sistema del ejemplo, el cambio es el siguiente:

```
/dev/hda:  
Timing buffer-cache reads: 128 MB in 1.01 seconds = 126.73 MB/sec  
Timing buffered disk reads: 64 MB in 16.53 seconds = 3.87 MB/sec
```

El rendimiento de la unidad de disco ha aumentado desde 3.71 MB/seg a 3.87 MB/seg. Ni mucho ni poco. Quizá su cambio sea parecido a éste. Posiblemente pueda hacerlo mejor si su unidad de disco y su controlador son realmente nuevos. Probablemente alcance de unos 20 a unos 30 MB/seg. Sea precavido, sin embargo, cuando curiose con `hdparam`, pues se pueden dañar sus datos, por eso, tal y como se mencionó antes, tiene que realizar una copia de sus datos antes de cambiar las opciones específicas de hardware del disco duro que acabamos de ver.

Si `hdparam` informa que el soporte I/O asignado es de 16-bit para su sistema y tiene un sistema de unidades de disco lo suficientemente nuevo (uno o dos años), debería intentar un soporte I/O de 32-bit. Puede asignarlo utilizando la opción `-c` en `hdparam`. Esta opción tiene tres valores:

0: Permite, por defecto, soporte I/O 16-bit.

1: Permite soporte de 32-bit.

3: Permite soporte de 32-bit con una secuencia de sincronización determinada que necesita muchos conjuntos de chips IDE/EIDE. Además, también es el valor que funciona en la mayoría de los sistemas.

Las opciones se fijan del siguiente modo:

```
/usr/local/sbin/hdparm -m16 -c3 /dev/hda
```

Observe que se utilizaron las opciones `-m16` y `-c3` para activar el soporte I/O de 32-bit. Al ejecutar el programa con la opción `-t` se muestran los siguientes resultados:

```
/dev/hda:  
Timing buffered disk reads: 64 MB in 8.96 seconds = 7.14 MB/sec
```

Como puede ver, prácticamente se ha doblado el rendimiento del sistema de discos. Sin embargo, debería ser capaz de aumentar el rendimiento incluso más. Por ejemplo, si su disco soporta acceso de memoria directo (DMA) debería ser capaz de utilizar la opción `-d`, que activa el modo DMA.

Normalmente, las opciones `-d1 -X32` y `-d1 -X66` se utilizan juntas para sacar partido de las capacidades DMA de sus sistema de discos. El primer conjunto de opciones (`-d1 -X32`) activa el DMA modo 2 multiword para el disco, y el siguiente conjunto de opciones (`-d1 -X66`) activa el UltraDMA modo 2 para discos que soportan la característica de coordinación de UltraDMA. Estas opciones pueden aumentar enormemente el rendimiento de su sistema. He visto medias de transferencia de 20 MB/seg con estas opciones en distintos discos EIDE/ATA nuevos.

Hay otra opción, `-ul`, que puede ser muy útil para el aumento general del rendimiento del sistema. Esta opción permite al disco desenmascarar otras interrupciones durante la interrupción de un disco, lo que significa que el sistema

operativo puede atender a otras interrupciones como las de I/O de red, I/O en serie, y similares, mientras esperan que finalice una transferencia de datos basada en el disco.

Hay muchas más opciones `hdparm` con las que puede experimentar; sin embargo, tenga cuidado con la mayoría de las opciones porque es muy fácil que se corrompan los datos. Además, una vez que encuentre un conjunto de opciones que funcionen, debería utilizarlas en el comando `hdparm` en el script `/etc/rc.d/rc.local`, de modo que se asignen cada vez que inicie el sistema. Por ejemplo, yo añadí la siguiente línea al archivo `/etc/rc.d/rc.local` en uno de mis sistemas Red Hat Linux nuevo:

```
hdparm -m16 -c3 -u1 -d1 -X66 /dev/hda
```

Ahora que su disco duro está ajustado para mejor rendimiento, vamos a ver cómo puede ajustar el sistema de archivos que actúa como interfaz para sus discos. Como Linux utiliza el sistema de archivos `ext2`, en la próxima sección vamos a ver los aspectos que hay que ajustar en ese sistema de archivos.

## Ajustar el sistema de archivos `ext2`

El sistema de archivos `ext2` ha sido durante años el sistema de archivos de Linux. No es el mejor sistema de archivos del mundo, pero funciona razonablemente bien. Una de las formas de aumentar el rendimiento del sistema de archivos `ext2` es cambiar el tamaño por defecto del bloque de 1024 a un múltiplo de 1024 (normalmente menos de 4096) para servidores con muchos archivos de gran tamaño. Vamos a ver cómo puede cambiar el tamaño del bloque.

### Cambiar el tamaño del bloque del sistema de archivos `ext2`

Para encontrar el tipo de archivos (en términos de tamaño) que tiene una partición `ext2` determinada, haga lo siguiente:

1. Establézcase como raíz y cambie el directorio de máximo nivel de la partición `ext2`.
2. Ejecute el siguiente comando, que es realmente un pequeño script que utiliza las utilidades `find` y `awk`. Este script de la línea de comando muestra todos los archivos y sus tamaños, y para terminar proporciona un tamaño medio y total de la partición completa.

```
find . -type f -exec ls -l {} \; | \ 
awk 'BEGIN {tsize=0;fcnt=1;} \
{ printf("%03d File: %s size: %d bytes\n",fcnt++, $9, \
$5); \
tsize += $5; } \
END { printf("Total size = %d\nAverage file size = %.02f\n", \
tsize, tsize/fcnt); }'
```

3. Una vez que conoce el tamaño medio de un sistema de archivos puede determinar si debería cambiar el tamaño del bloque. Imagine que el tamaño medio de sus archivos es de 8192, que es  $2 \times 4096$ . Puede marcar un tamaño de bloque de 4096.
4. No puede cambiar el tamaño del bloque de un sistema de archivos ext2 que ya existe sin reconstruirlo. Por lo tanto, tiene que realizar un backup de todos sus archivos de sistema de archivos u reconstruirlo utilizando el comando `/sbin/mke2fs /dev/partition -b 4096`. Por ejemplo, si ha copiado la partición `/dev/hda7` y quiere cambiar el tamaño del bloque a 4096, utilice el comando `/sbin/mke2fs /dev/hda7 -b 4096`.

**NOTA:** Cambiar el tamaño del bloque a un número superior al número por defecto (1024) dará lugar a un gran aumento de rendimiento, como resultado de una reducción en el número de búsquedas, además de a una sesión fsck potencialmente más rápida durante el arranque, menor fragmentación de archivos, y a otros efectos. Sin embargo, aumentar el tamaño del bloque a ciegas (es decir, sin conocer el tamaño medio de los archivos) puede dar lugar a un gran gasto de espacio. Si el tamaño medio de los archivos es de 2010 bytes en un sistema con bloques de 4096-byte, cada archivo gastará una media de 2086 bytes ( $4096 - 2010$ ). Por lo que ha de conocer el tamaño de sus archivos antes de cambiar el tamaño del bloque.

## Instalar e2fsprogs para ajustar el sistema de archivos ext2

Para ajustar el sistema de archivos ext2, tiene que instalar el paquete de utilidades e2fsprogs del siguiente modo:

1. Baje la distribución fuente del programa `e2fsprogs-version.src.rpm` (reemplace `version` con el número de la última versión) de [www.rpmfind.net](http://www.rpmfind.net). Yo bajé el paquete `e2fsprogs-1.19-0.src.rpm`. También puede obtener la fuente del sitio del proyecto e2fsprogs en <http://e2fsprogs.sourceforge.net.su>.
2. Ejecute el comando `rpm -ivh e2fsprogs-version.src.rpm` para extraer la fuente en el directorio `/usr/src/redhat/SOURCES/`. La fuente RPM lanza un archivo `e2fsprogs-version.tar.gz`, que ha de extraerse con el comando `tar xvzf e2fsprogs-version.tar.gz`. Esto da lugar a un subdirectorio llamado `e2fsprogs-version`.
3. Cambie al nuevo subdirectorio `e2fsprogs-version`.
4. Ejecute `mkdir build` para crear un subdirectorio nuevo y entonces cambie a ese subdirectorio.

- Ejecute el script `../configure` para configurar el árbol fuente. Ejecute, entonces, la utilidad `make` para crear los binarios. Ejecute `make check` para garantizar que todo está construido correctamente. Finalmente, ejecute el comando `make install` para instalar los binarios.

Una vez que tiene instaladas las utilidades `e2fsprogs` puede empezar a utilizarlas tal y como se discute en la siguiente sección.

## **Utilizar la utilidad `tune2fs` para la puesta a punto del sistema de archivos**

Puede utilizar las utilidades `tune2fs` para ajustar varios aspectos de un sistema de archivos `ext2`. Sin embargo, nunca debe aplicar las utilidades `ext2` en un `ext2` montado y debe realizar siempre un backup de sus datos cada vez que modifique cualquier cosa que pertenezca al sistema. En esta sección, vamos a ver cómo emplear la utilidad `tune2fs` (que forma parte del paquete `e2fsprogs`) para poner a punto un sistema de archivos `ext2` sin montar llamado `/dev/hda7`. Si utiliza uno o más de los aspectos que se discuten a continuación, no olvide cambiar el nombre de la partición (`/dev/hda7`) con el nombre apropiado. Para empezar, ejecute el comando siguiente para determinar qué `tune2fs` muestra aspectos actuales para el `/dev/hda7` sin montar:

```
/sbin/tune2fs -l /dev/hda7
```

La salida será algo parecido a lo que tenemos a continuación:

```
tune2fs 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95/08/09
Filesystem volume name: <none>
Last mounted on: <not available>
Filesystem UUID: 5d06c65b-dd11-4df4-9230-a10f2da783f8
Filesystem magic number: 0xEF53
Filesystem revision #: 1 (dynamic)
Filesystem features: filetype sparse_super
Filesystem state: clean
Errors behavior: Continue
Filesystem OS type: Linux
Inode count: 1684480
Block count: 13470471
Reserved block count: 673523
Free blocks: 13225778
Free inodes: 1674469
First block: 1
Block size: 1024
Fragment size: 1024
Blocks per group: 8192
Fragments per group: 8192
Inodes per group: 1024
Inode blocks per group: 128
Last mount time: Thu Feb 15 17:51:19 2001
Last write time: Thu Feb 15 17:51:51 2001
```

Mount count:	1
Maximum mount count:	20
Last checked:	Thu Feb 15 17:50:23 2001
Check interval:	15552000 (6 months)
Next check after:	Tue Aug 14 18:50:23 2001
Reserved blocks uid:	0 (user root)
Reserved blocks gid:	0 (group root)
First inode:	11
Inode size:	128

Los aspectos que se discuten a continuación son los que están marcados en negrita en el listado anterior:

- **Error behavior:** dicta cómo se comporta el kernel cuando se detectan errores en el sistema de archivos. Hay tres valores posibles: **continue**, **remount-ro** (sólo lectura), **panic**. La asignación por defecto es que la ejecución continúe aunque haya un error.
- **Mount count:** el tiempo que tarda en montar este sistema de archivos.
- **Maximum mount count:** significa que después de que el máximo número de modos de lectura / escritura monte el archivo, éste estará sujeto a una sesión de revisión **fsck**, durante el próximo ciclo de arranque.
- **Last checked:** muestra la fecha en la que se realizó una revisión **fsck** por última vez. El intervalo de comprobación entre dos sesiones **fsck** consecutivas.
- **Check interval:** sólo se utiliza si no se alcanza el máximo número de montajes de lectura / escritura durante el intervalo. En otras palabras, si no ha desmontado el sistema de archivos durante 6 meses, entonces, incluso aunque el número de montajes fuese sólo 2, se forzaría una revisión **fsck**, porque el sistema de archivos habría superado el intervalo de comprobación.
- **next check after:** la próxima fecha de revisión **fsck**.
- **reserved block UID** y **reserved block GID**: muestran el usuario y el grupo propietarios de la porción reservada de este sistema de archivos. Por defecto, la porción reservada es para que la utilice la raíz (UID = 0, GID = 0).

En un sistema de archivos sin montar como `/dev/hda7`, puede cambiar el número máximo de montajes de lectura / escritura a un número más adecuado para sus necesidades, utilizando la opción **-c** con `tune2fs`. Por ejemplo, `/sbin/tune2fs -c 1 /dev/hda7` forzará a que `fsck` compruebe el sistema de archivos cada vez que `fsck` esté basado en el tiempo.

Por ejemplo, el comando `/sbin/tune2fs --i17d /dev/hda7` garantiza que se imponga la comprobación `fsck` si el sistema de archivos se vuelve a

montar en modo lectura/ escritura tras una semana. Del mismo modo, el comando /sbin/tune2fs --i0 /dev/hda7 desactiva las comprobaciones fsck basadas en el tiempo.

## Comprobar y reparar un sistema de archivos ext2 con e2fsck

En el caso de que tenga un sistema de archivos ext2 corrompido, puede utilizar la utilidad e2fsck para tratar de arreglarlo. Para comprobar una partición utilizando e2fsck, debe desmontarla primero y ejecutar el comando /sbin/e2fsck /dev/device en el que /dev/device es su disco duro. Por ejemplo, para forzar una verificación fsck en un dispositivo llamado /dev/hda7, utilice el comando /sbin/e2fsck -f /dev/hda7. Esta verificación mostrará una salida como la que se muestra a continuación:

```
e2fsck 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95/08/09
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/hda7: 12/1684256 files (0.0% non-contiguous), 52897/
3367617 blocks
```

La utilidad e2fsck le hará una serie de preguntas, que puede obviar utilizando la opción -p.

## Aumentar la seguridad con un sistema de archivos journaling para Linux

El último grito en los sistemas de archivos Linux son los sistemas de archivos journaling. Un sistema de archivos journaling es, sencillamente, un sistema de archivos basado en transacciones. Cada actividad del sistema de archivos que cambia el sistema, se registra en un registro de transacción. Si tiene lugar un incidente, el sistema de archivos puede repetir las transacciones necesarias para volver al estado estable en poco tiempo. Es una técnica que utilizan muchas ingenierías de bases de datos, como IBM DB2 y Oracle, para garantizar que el sistema de archivos tenga siempre un estado conocido y recuperable.

El problema con el sistema de archivos ext2 es que si tiene lugar un incidente, el sistema de archivos se puede quedar en tal estado que se podría corromper más allá de cualquier recuperación. El programa fsck, que se utiliza para comprobar y reparar el sistema de archivos, a menudo no puede hacer demasiado para arreglar este tipo de problemas. Con un sistema de archivos journaling, este tipo de pesadillas son cosa del pasado. Como los registros de transacción, registran todas las actividades de un sistema de archivos, la recuperación de un incidente es más rápida y la pérdida de datos es mínima.

**NOTA:** Un sistema de archivos journaling no registra datos en el archivo de registros; simplemente registra meta datos relacionados con las operaciones de discos, de modo que repetir el registro sólo da lugar a un sistema de archivos coherente con la relación estructural y con el reparto de recursos. Por eso, es posible que tenga lugar una pequeña pérdida de datos. Además, el registro es objeto de errores del medio al igual que ocurre en cualquier otra actividad. Por eso, si los medios son malos, el sistema de archivos journaling no va a ayudar mucho.

El sistema de archivos journaling es nuevo en Linux pero ya estaba disponible en otras plataformas. Aunque el soporte del sistema de archivos es nuevo en Linux, ha recibido mucho interés por parte de industrias interesadas en utilizar Linux en la empresa; por eso, el sistema de archivos journaling madurará rápidamente. Le recomiendo que comience utilizando el sistema de archivos journaling en un nivel experimental y que se vaya acostumbrando a sus características.

Hay varias versiones del sistema de archivos journaling disponibles. ReiserFS se discute en la siguiente sección porque está incluida con el kernel 2.4.1 (o superiores) de Linux, pero sería un descuido por mi parte no ofrecer una lista de los sistemas de archivos journaling disponibles principales. Estos son:

- **JFS de IBM, de código fuente abierto, para Linux.** JFS se ha transportado desde AIX, la propia plataforma del sistema operativo de IBM, pero sigue sin estar preparada para su uso en producción. Puede encontrar más información en <http://oss.software.ibm.com/developerworksopensource/jfs>.
- **El sistema de archivos ext3 de Red Hat, con capacidades journaling.** Tampoco está preparado para su uso en producción. Puede bajar la versión alfa de ext3 en <ftp://ftp.linux.org.uk/pub/linux/sct/fs/jfs/>.
- **ReiserFS, desarrollado por Namesys, que está incluido en la distribución fuente del kernel de Linux.** Se utiliza más que el resto de los sistemas de archivos para Linux. Hasta ahora, aparece como el principal sistema de archivos journaling para Linux. ReiserFS fue desarrollado por Hans Reiser, el cual garantizó un desarrollo consolidado para compañías comerciales como MP3, BigStorage.com, SuSe y Ecila.com. Estas compañías necesitaban sistemas de archivos flexibles y mejores, y se dirigieron rápidamente a la experiencia de usuario beta. Puede encontrar más información en ReiserFS en [www.namesys.com](http://www.namesys.com).

**ADVERTENCIA:** En el momento en el que se escribió esto, el sistema de archivos ReiserFS no se podía utilizar con NFS sin parches, el cual no está disponible oficialmente para el kernel 2.4.1 o superior.

- **Sistema de archivos XFS, desarrollado por Silicon Graphics, Inc. (SGI).** Puede encontrar más información sobre XFS en <http://oss.sgi.com/projects/xfs/>. XFS es un sistema de archivos de 64-bit rápido y sólido, lo que significa que puede soportar archivos muy grandes (9 millones de terabytes) e incluso sistemas de archivos muy grandes (18 millones de terabytes).

## Compilar e instalar ReiserFS

Antes de poder utilizar ReiserFS, tiene que compilarlo e instalarlo. Siga estos pasos para compilar e instalar soporte ReiserFS (reiserfs) en el kernel 2.4.1 de Linux o en uno superior:

1. Obtenga la última fuente del kernel de Linux de [www.kernel.org](http://www.kernel.org) y extráigalo en el directorio /usr/src/linux-version como raíz, donde version es la versión actual del kernel. Los siguientes pasos suponen que tiene el kernel 2.4.1.
2. Ejecute make menuconfig desde el directorio /usr/src/linux-2.4.1.
3. Seleccione el submenú Code maturity level options y, utilizando la barra espaciadora, seleccione Prompt for development and/or incomplete code/drivers. Salga del submenú.
4. Seleccione el submenú File systems. Utilizando la barra espaciadora, seleccione Reiserfs support para incluirlo como un módulo kernel y salga del submenú.

**NOTA:** No elija la opción Have reiserfs do extra internal checking bajo la opción ReiserFS support. Si asigna yes, entonces reiserfs realizará una amplia comprobación de la lógica de esta operación, lo que hará que sea muy lento.

5. Seleccione también las otras características kernel que utilice.
6. Salga del menú principal y guarde la configuración del kernel.
7. Ejecute el comando make dep tal y como se indica en el programa menuconfig.
8. Ejecute make bzImage para crear el nuevo kernel. Entonces ejecute make y make modules\_install para instalar los módulos nuevos en las localizaciones apropiadas.
9. Cambie el directorio a arch/i386/boot. Si la arquitectura de su hardware es Intel, tiene que reemplazar i386 y posiblemente necesite más

instrucciones sobre el kernel, que puede encontrar en manuales, para compilar e instalar su versión del kernel. Supongo que la mayoría de los sistemas de los lectores están basados en i386.

10. Copie bzImage en /boot/vmlinuz-2.4.1 y edite el archivo /etc/lilo.conf para incluir una nueva configuración como la siguiente:

```
image=/boot/vmlinuz-2.4.1
      label=linux2
      read-only
      root=/dev/hda1
```

11. Ejecute el comando /sbin/lilo para volver a configurar lilo y reinicie su sistema. Introduzca linux2 en el prompt lilo e inicie el kernel nuevo. Si tiene algún problema, debería ser capaz de reiniciar su kernel estándar de Linux, que debería aparecer por defecto automáticamente. Una vez que ha iniciado el nuevo kernel, está preparado para utilizar ReiserFS (reiserfs).

## Montar el sistema de archivos ReiserFS

Como ReiserFS (reiserfs) sigue en la categoría de "experimental", le recomiendo que lo utilice en una parte de su sistema que no sea demasiado crítica. De forma ideal, debería dedicar un disco completo o una o más particiones para ReiserFS de modo que pudiese utilizarlo de forma segura y para ver si le gusta.

Para utilizar ReiserFS con una nueva partición llamada /dev/hda7, haga lo siguiente:

1. Como raíz, garantice que la partición está asignada como Linux native (83) utilizando fdisk u otra herramienta de partición de discos.
2. Tiene que crear un sistema de archivos ReiserFS (reiserfs) en la nueva partición, utilizando el comando /sbin/mkreiserfs /dev/hda7.
3. A continuación, tiene que crear un directorio o un punto de montaje para el nuevo sistema de archivos. Por ejemplo, yo he creado un punto de montaje llamado /jfs utilizando el comando mkdir /jfs.
4. Finalmente, monte el sistema de archivos utilizando el comando mount -t reiserfs /dev/hda7 /jfs. Ahora puede acceder a él desde el punto de montaje /jfs.

Para ver cómo funciona un sistema de archivos journaling frente a un sistema de archivos ext2, puede utilizar un benchmark en la sección siguiente.

## Utilizar un benchmark para ReiserFS

Voy a suponer que ha creado un sistema de archivos ReiserFS en /dev/hda7 y que lo ha montado en /jfs.

**ADVERTENCIA:** Para realizar esta comparación, no debe almacenar ningún dato en esta partición. Por eso, tiene que realizar un backup de todo lo que tenga en /jfs ya que se va a borrar todo lo que se encuentre en /jfs durante este proceso.

Para realizar una comparación con benchmark ReiserFS, siga los siguientes pasos:

1. Tiene que crear un script shell llamado reiserfs\_vs\_ext2.bash en el directorio /tmp. Este script se puede observar en el listado 23.1.

**Listado 23.1.** /tmp/reiserfs\_vs\_ext2.bash

```
#!/bin/bash
#
# Este script está creado basado en el script file_test
# del benchmark de http://www.namesys.com
#
if [ $# -lt 6 ]
then
        echo Usage: file_test dir_name device nfiles size1 size2
        log_name
        exit
fi

TESTDIR=$1
DEVICE=$2
LOGFILE=$6

/bin/umount $TESTDIR
/sbin/mkreiserfs $DEVICE
mount -t reiserfs $DEVICE $TESTDIR

echo 1. reiserfs 4KB creating files ...
echo "reiserfs 4KB create" $3 "files of size: from " $4 "to" $5
> $LOGFILE
(time -p ./mkfile $TESTDIR $3 $4 $5)>> $LOGFILE 2>&1
echo done.
sync
df >> $LOGFILE

/bin/umount $TESTDIR
/sbin/mke2fs $DEVICE -b 4096
mount -t ext2 $DEVICE $TESTDIR

echo 2. ext2fs 4KB creating files ...
echo "ext2fs 4KB create" $3 "files of size: from " $4 "to" $5
>> $LOGFILE
```

```
(time -p ./mkfile $TESTDIR $3 $4 $5)>> $LOGFILE 2>&1
echo done.
sync
df >> $LOGFILE

/bin/umount $TESTDIR
```

2. Además, tiene que bajar un pequeño programa C llamado `mkfile.c`, desarrollado por el equipo ReiserFS, a `/tmp`, desde [www.namesys.com/filetest/mkfile.c](http://www.namesys.com/filetest/mkfile.c).
3. Desde el directorio `/tmp`, compile `mkfile.c`, utilizando el comando `gcc -o mkfile mkfile.c`.
4. Cambie el permiso de los programas `reiserfs_vs_ext2.bash` y `mkfile` utilizando el comando `chmod 755 reiserfs_vs_ext2.bash mkfile`.
5. Ejecute el comando siguiente desde el directorio `/tmp` como raíz:  
`./reiserfs_vs_ext2.bash /jfs /dev/hda7 100000 1024 4096 log`
6. Le pedirán que confirme que realmente quiere perder todos los datos de `/dev/hda7`. Como ya ha vaciado esta partición para probar, diga que sí y continúe. Esta prueba creará 100.000 archivos de distintos tamaños, desde 1K a 4K en los sistemas de archivos ReiserFS (`reiserfs`) y ext2, creando cada uno de estos dos sistemas de archivos en `/dev/hda7`. Los resultados se registrarán en el archivo `/tmp/log`. A continuación tiene una muestra del archivo `/tmp/log`:

```
reiserfs 4KB create 100000 files of size: from 1024 to 4096
real 338.68
user 2.83
sys 227.83
Filesystem      1k-blocks      Used Available Use%
Mounted on
/dev/hda1        1035660     135600    847452   14% /
/dev/hda5        4134868    2318896   1605928   60% /usr
/dev/hda7        13470048   332940   13137108   3% /jfs
ext2fs 4KB create 100000 files of size: from 1024 to 4096
real 3230.40
user 2.87
sys 3119.12
Filesystem      1k-blocks      Used Available Use%
Mounted on
/dev/hda1        1035660     135608    847444   14% /
/dev/hda5        4134868    2318896   1605928   60% /usr
/dev/hda7        13259032    401584   12183928   4% /jfs
```

Observe que para crear archivos de tamaño 1K a 4K, ReiserFS (`reiserfs`) tarda 338,68 segundos de tiempo real; ext2 tarda 3230,40 segundos. De modo que el rendimiento de ReiserFS (`reiserfs`) es muy bueno.

# Compartir espacio de disco con el servidor NFS

Aunque puede compartir contenido estático mediante un sistema de archivos de red (NFS), le recomiendo utilizar copias locales de los archivos estáticos. Los archivos almacenados en un disco local darán lugar a un acceso mucho más rápido que cualquier solución de almacenaje de red a no ser que elija una solución SAN de alta calidad.

Le recomiendo colocar en una partición NFS, únicamente los archivos que quiera compartir, que cambien frecuentemente, o que deban utilizar los mismos datos a lo largo de todos los nodos Web. Por ejemplo, sus aplicaciones Web pueden compartirse en todos los nodos, de modo que tenga una sola copia de todo el software. Sin embargo, es importante que se comparten los datos de estas aplicaciones. Por ejemplo, si tiene un script CGI que escribe datos en una sesión de disco para un carrito de compra, cuando el usuario realiza la siguiente solicitud, el esquema de balance de carga que esté utilizando, dirigirá al usuario a un servidor Web distinto de su red, lo que significa que el dato de sesión no está disponible para los otros servidores Web o que el usuario tendrá que reiniciar su carrito de la compra.

Tiene que crear un directorio en el que guarde todos los datos que se deban compartir. Por ejemplo, puede crear `/www/cgi-data` como el directorio de datos del script CGI en el servidor NFS. Puede crear subdirectorios para cada aplicación CGI y configurar cada aplicación para que escriba en su propio directorio dentro de `/www/cgi-data`. Entonces, montar este directorio mediante NFS lo dejará disponible a todos los servidores Web de la red. La siguiente sección le muestra cómo hacerlo.

## Establecer un servidor NFS

Un servidor NFS tiene que ejecutar un programa llamado portmapper (también llamado portmap o rpc.portmap), que es iniciado normalmente por un script `rc`. Para comprobar si portmapper se está ejecutando ya, utilice el siguiente comando:

```
ps auxw | grep portmap
```

Bajo RedHat Linux, el script `/etc/rc.d/rc3.d/S40portmap` (es decir, el script `/etc/rc.d/init.d/portmap.init`) inicia el programa portmapper automáticamente, de modo que no hay que iniciararlo manualmente.

El siguiente paso es modificar el archivo `/etc/exports` para decirle al sistema qué sistema de archivos o directorios tiene que exportar a los clientes NFS. Como XC News sólo necesita exportar el directorio `/www/cgi-data` a los servidores Web, el archivo exportado en el host `ns.xcnews-lan.com` es como este:

```
/www/cgi-data    www1.xcnews-lan.com(rw)    www2.xcnews-lan.com(rw)
```

Esta línea le dice al servidor NFS que permita a www1.xnews-lan.com y a www2.xcnews-lan.com acceso de lectura y escritura en el directorio /www/cgi-data.

**NOTA:** La sintaxis para los archivos exportados no debe ser la misma para todas las marcas de Unix.

Los programas que deben ejecutarse a continuación, son mountd (rpc.mountd) y nfsd (rpc.nfsd). Estos dos programas se inician también automáticamente desde los scripts rc en /etc/rc.d/rc3.d. Cada vez que se realiza un cambio en el archivo /etc(exports, sin embargo, hay que informar a estos dos programas sobre el cambio. Un script llamado exportfs puede reiniciar estos dos programas, del siguiente modo:

```
exportfs
```

Si no se encuentra exportfs en un sistema, entonces se puede utilizar un script como el siguiente:

```
#!/bin/sh
killall -HUP /usr/sbin/rpc.mountd
killall -HUP /usr/sbin/rpc.nfsd
echo re-exported file systems
```

Este script utiliza el programa killall que se encuentra en la mayoría de los sistemas Linux; si no está disponible, siempre puede ejecutar un comando ps, que se encuentra en PID para estos procesos, y realizar manualmente un kill -HUP para cada proceso. Ahora tiene que asegurarse de que tanto mountd como nfsd se están ejecutando adecuadamente, ejecute un programa llamado rpcinfo, del siguiente modo:

```
rpcinfo -p
```

La salida será como esta:

program	vers	proto	port	
100000	2	tcp	111	rpcbind
100000	2	udp	111	rpcbind
100005	1	udp	635	mountd
100005	2	udp	635	mountd
100005	1	tcp	635	mountd
100005	2	tcp	635	mountd
100003	2	udp	2049	nfs
100003	2	tcp	2049	nfs

Esto muestra que portmapper, mountd y nfsd tienen declarados sus servicios y que funcionan bien. Antes de establecer el lado del cliente de NFS en los servido-

res Web, es importante garantizar que los aspectos de seguridad están dirigidos, tal y como se discute en la siguiente sección.

## Aspectos de seguridad del servidor

Se puede engañar al portmapper, en combinación con nfsd, haciendo posible obtener archivos en los servidores NFS sin privilegios. El portmapper Linux es relativamente seguro frente a ataques, y se puede hacer más seguro añadiendo la siguiente línea en el archivo /etc/hosts.deny:

```
portmap: ALL
```

El sistema rechazará el acceso a portmapper a todo el mundo. Ahora el archivo /etc/hosts.allow tiene que modificarse del siguiente modo:

```
portmap: 192.168.1.0/255.255.255.0.
```

Esto permite que todos los host de la red 192.168.1.0 tengan acceso a los programas administrados por portmapper, como nfsd y mountd.

**ADVERTENCIA:** Nunca utilice nombres de host en la línea portmap en /etc/hosts.allow porque puede dar lugar a un ataque informático a la seguridad portmap, lo que provocará búsquedas de nombres de host en un bucle.

Otro aspecto de seguridad del lado del servidor es si se permite que la cuenta root de un cliente, sea tratada como root en un servidor. Por defecto, Linux prohíbe root en el lado del cliente de NFS sea tratado como root del lado del servidor. En otras palabras, un archivo exportado propiedad de root en el servidor no se puede modificar por usuario root del cliente. Para forzar esta regla explícitamente, se puede modificar el archivo /etc(exports:

```
/www/cgi-data      www1.xcnews-lan.com(rw,  root_squash)  
www2.xcnews-lan.com(rw,  root_squash)
```

Ahora, si un usuario con UID 0 (usuario root) en el cliente, intenta acceder (leer, escribir o eliminar) al sistema de archivos, el servidor sustituye la UID de la cuenta "nobody" del servidor. Esto significa que el usuario root del cliente no puede acceder o cambiar archivos a los que sólo puede acceder o que solo puede cambiar el root del servidor.

**TRUCO:** Para conceder acceso root a un sistema de archivos NFS, incluye la opción no\_root\_squash.

En este momento, el servidor NFS está preparado y seguro, de modo que vamos a preparar a continuación los hosts clientes NFS.

## Establecer un cliente NFS

Por defecto, Red Hat Linux soporta sistemas de archivos NFS, de modo que no hay necesidad de utilizar el kernel. Para montar el directorio /www/cgi-data exportado por el host ns.xcnews-lan.com, añada la siguiente línea al archivo /etc/fstab para ambos servidores Web:

```
ns.xcnews-lan.com:/www/cgi-data      /www/cgi-data      nfs
```

Esta línea monta automáticamente el directorio /www/cgi-data cuando se vuelven a iniciar los servidores Web.

A continuación, tiene que crear el directorio /www/cgi-bin en ambos sistemas y montar manualmente el directorio utilizando el comando mount, del siguiente modo:

```
ns.xcnews-lan.com:/www/cgi-data      /www/cgi-data      nfs
```

**ADVERTENCIA:** Un problema típico en el montaje de NFS tiene lugar porque muchos desarrolladores olvidan ejecutar exportfs (es decir, restart rpc.mountd y rpc.nfssd) tras modificar el archivo /etc(exports en el servidor NFS).

Desmontar un sistema de archivos NFS es exactamente lo mismo que desmontar un sistema de archivos local. Observe que también es posible incrementar la seguridad cliente NFS sin confiar demasiado en el servidor NFS. Por ejemplo, puede desactivar programas suid (con privilegios de root) para desahogar al sistema de archivos NFS con una opción nosuid. Esto significa que el usuario root del servidor no puede realizar un programa suid-root en el sistema de archivos, registrarse en el cliente como un usuario normal, y entonces utilizar el programa suid-root para convertirse en el root del cliente. Además, es posible prohibir la ejecución de archivos en el sistema de archivos montado con la opción noexec. Puede introducir estas opciones en la columna de opciones de la línea que describe su punto de montaje NFS en el archivo /etc/fstab.

En este momento, están preparados tanto el esquema de distribución de archivos como el directorio de datos CGI basado en NFS. Es el momento de configurar Apache y de garantizar su seguridad.

El principal cuello de botella en un entorno NFS, es la velocidad del disco I/O del servidor NFS. La velocidad del disco I/O depende del tipo de sistema de discos que esté utilizando con su servidor NFS. Por ejemplo, si utiliza discos IDE en un servidor NFS, no va a tener un rendimiento muy alto si se compara con unidades de disco ultrawide SCSI, las cuales tienen altas cotas de RPM. El máximo número de operaciones I/O por segundo, marcará el funcionamiento del servidor NFS. Yo he utilizado un sistema Intel Xeon 500 con 10 discos ultrawide SCSI, en un RAID 5 como servidor NFS para, aproximadamente, 50 usuarios, con gran éxito.

Una vez que se ha decidido por un buen sistema de discos como un RAID 5 utilizando un array de 10K RPM de discos ultrawide SCSI, con un controlador de discos que tenga un gran caché integrado, su siguiente cuello de botella es la propia red. Un buen modo de reducir la pérdida de rendimiento es aislar el tráfico de alto ancho de banda en su propia red. Por eso, recomiendo que conecte su servidor o sus servidores NFS a sus clientes NFS utilizando una Ethernet de 100Mbits dedicada. En otras palabras, tiene que crear una backbone (red troncal) NFS que sólo trabaje con paquetes NFS. Esto dará lugar a una red NSF de alto rendimiento. Las opciones de configuración del software que pueden ayudarle a ajustar su servidor NFS, se comentan en las secciones siguientes.

## **Optimizar el tamaño del bloque de caracteres de lectura / escritura**

El tamaño por defecto del bloque de caracteres de lectura y escritura para NFS es de 4096 bytes (4K), que podría no ser el óptimo para todas las situaciones. Puede realizar una prueba para determinar si cambiando el tamaño del bloque aumentará el rendimiento. A continuación tenemos el modo de realizar esta prueba. Esta prueba supone que tiene un servidor NFS ejecutándose en un sistema Linux y que también tiene un sistema cliente NFS basado en Linux. La prueba también supone que el cliente ha montado un sistema de archivos llamado /mnt/nfs1 desde el servidor NFS.

1. Establézcase como root en la máquina NFS cliente.
2. Para empezar, necesita saber la cantidad total de memoria que tiene su sistema. Debería conocer el valor porque es su sistema, pero si no lo recuerda, puede ejecutar el comando `cat /proc/meminfo` para ver la información sobre la memoria de su sistema. Esto da lugar una salida parecida a la siguiente:

```
total: used: free: shared: buffers: cached:  
Mem: 263720960 260456448 3264512 30531584 228245504  
6463488  
Swap: 271392768 6209536 265183232  
MemTotal: 257540 kB  
MemFree: 3188 kB  
MemShared: 29816 kB  
Buffers: 222896 kB  
Cached: 6312 kB  
BigTotal: 0 kB  
BigFree: 0 kB  
SwapTotal: 265032 kB  
SwapFree: 258968 kB
```

3. La cantidad total de memoria del sistema que se muestra bajo la cabecera de la columna `total:`; divide este número entre 1.048.576 (1024x1024) para obtener (aproximadamente) el tamaño total de la memoria en

megabytes. En el ejemplo anterior, este número es 251MB. La mayoría de los BIOS de los sistemas PC no informan exactamente de la memoria total, por lo que siempre tendremos un número aproximado basado en lo que sepamos sobre la cantidad de memoria total. En mi ejemplo, yo sé que el sistema debería tener 256MB de RAM, por lo que utilizaré 256MB como tamaño de la memoria para la prueba.

4. A continuación, cambie el directorio al archivo NSF recién montado, /mnt/nfs1. Ejecute el comando du para comprobar si tiene al menos 512MB (2 x total RAM) de espacio libre disponible en el directorio NFS. Si no lo hace, no podrá continuar con esta prueba.
5. La siguiente tarea es medir el rendimiento de escritura de su sistema NFS, escribiendo un archivo de 512MB (16K/bloque x 32,768 bloques) llamado 512MB.dat en el directorio /mnt/nfs1, utilizando el comando siguiente:

```
time dd if=/dev/zero \
    of=/mnt/nfs1/512MB.dat \
    bs=16k count=32768
```

Este comando ejecuta el comando time, el cual registra el tiempo de ejecución que necesita el programa como primer argumento. En este caso, se ha controlado el tiempo del comando dd. El comando dd da lugar a un archivo input (utilizando la opción if) llamado /dev/zero. Este archivo es un dispositivo especial que devuelve el carácter 0 (cero) cuando lee. En otras palabras, si abre este archivo para lectura, devolverá un carácter 0 hasta que cierre el archivo. Esto nos ofrece un recurso sencillo para llenar un archivo output (que se especifica utilizando la opción of) llamado /mnt/nfs1/512MB.dat; le pedimos al comando dd que utilice un tamaño de bloque (que se especifica utilizando la opción bs) de 16K y que escriba un total de 32.768 bloques (que se especifica utilizando la opción count). Como 16K/bloque corresponden a 32,768 bloques de 512MB, tiene que crear dicho archivo. Una vez que se ejecuta este comando, imprimirá unas cuantas líneas como las siguientes:

```
32768+0 records in
32768+0 records out
1.610u 71.800s 1:58.91 61.7% 0+0k 0+0io 202pf+0w
```

Aquí, el comando dd ha leído 32.768 registros del dispositivo /dev/zero y, además, ha respondido escribiendo el mismo número de registros en el archivo /mnt/nfs1/512MB.dat file. La tercera línea indica que la operación de copia ha tardado 1 minuto y 58,91 segundos. Escriba esta línea en un archivo de texto del siguiente modo:

```
Write, 1, 1.610u, 71.800s, 1:58.91, 61.7%
```

Aquí puede observar que este fue el primer experimento (1) de escritura.

6. Para medir el rendimiento de lectura de su sistema NFS, puede leer simplemente el archivo 512MB que creó en el paso 5 y ver el tamaño que tiene. Para volver a leerlo y calcular el tiempo de acceso a la lectura, ejecute el siguiente comando:

```
time dd if=/mnt/nfs1/512MB.dat \
    of=/dev/null \
    bs=16k count=32768
```

Aquí, se calcula de nuevo el comando dd para leer el archivo /mnt/nfs1/512MB.dat como una entrada y salida del contenido de un archivo en /dev/null, que es el pozo sin fondo oficial para Linux. Al igual que antes, debería registrar el tiempo utilizado en el mismo archivo en el que respondió escribiendo el registro de rendimiento de lectura. Así por ejemplo, la prueba de lectura que utilizaba el comando anterior, mostraba la salida siguiente en mi sistema.

Registra la tercera línea del siguiente modo:

```
Read, 1, 1.970u, 38.970s, 2:10.44, 31.3%
```

Aquí puede observar que esta era la primera (1) experiencia de lectura.

7. Ahora elimine el archivo 512MB.dat de /mnt/nfs1 y desmonte la partición utilizando el comando umount /mnt/nfs1. Al desmontar el directorio NFS garantizamos que el caching de discos no influye en nuestro próximo conjunto de pruebas.
8. Repita la prueba de escritura y escritura (pasos del 5 al 7) al menos cinco veces. Debería tener un conjunto de notas del siguiente tipo:

```
Read, 1, 1.971u, 38.970s, 2:10.44, 31.3%
Read, 2, 1.973u, 38.970s, 2:10.49, 31.3%
Read, 3, 1.978u, 38.971s, 2:10.49, 31.3%
Read, 4, 1.978u, 38.971s, 2:10.49, 31.3%
Read, 5, 1.978u, 38.971s, 2:10.49, 31.3%

Write, 1, 1.610u, 71.800s, 1:58.91, 61.7%
Write, 2, 1.610u, 71.801s, 1:58.92, 61.7%
Write, 3, 1.610u, 71.801s, 1:58.92, 61.7%
Write, 4, 1.610u, 71.801s, 1:58.92, 61.7%
Write, 5, 1.611u, 71.809s, 1:58.92, 61.7%
```

9. Ahora, calcule la media del tiempo de lectura y escritura para la quinta columna (mostrada en negrita).

Hemos completado la primera fase de esta prueba. Hemos calculado la media de tiempo de acceso a lectura y a escritura para un archivo de 512MB. Para la segunda fase de la prueba, tiene que seguir los pasos siguientes:

1. Desmonte el directorio /mnt/nfs1 en el sistema NFS cliente, utilizando el comando umount /mnt/nfs1.

2. Modifique el archivo `/etc/fstab` en el sistema NFS cliente, de modo que el sistema de archivos `/mnt/nfs1` esté montado con las opciones `rsize=8192` y `wsize=8192`, tal y como se muestra a continuación.

```
nfs-server-host:/nfs1 /mnt/nfs1 nfs \
rsize=8192, wsize=8192 0 0
```

3. Vuelva a montar el directorio `/mnt/nfs1` utilizando el comando `mount /mnt/nfs1`.
4. Realice los pasos del 4 al 9 de la primera fase del experimento.
5. Compare las medias de acceso de lectura y escritura de las fases 1 y 2 de la prueba. Si el resultado de la fase 2 de la prueba es mejor, entonces el cambio de los bloques de lectura y escritura da lugar a un aumento del rendimiento en su NFS. En caso contrario, elimine las opciones `rsize=8192` y `wsize=8192` de `/etc/fstab`. Lo más probable es que el cambio en el tamaño del bloque de lectura y escritura, de lugar a un aumento de rendimiento en NFS. También puede experimentar con otros tamaños de bloque. Es aconsejable que utilice múltiplos de 1024 para el tamaño de los bloques, porque 1024 es el tamaño actual del bloque del sistema de archivos. Además, no debe utilizar números superiores a 8192 bytes. Si el cambio en el tamaño del bloque le funciona, mantenga `rsize=8192` y `wsize=8192` (o lo que considere óptimo tras la experimentación), en la línea `/etc/fstab` para la definición `/mnt/nfs1`.

## **Establecer la unidad de transmisión máxima apropiada**

El valor de la unidad de transmisión máxima (Maximum Transmission Unit, MTU), determina el tamaño que puede tener la transmisión de un solo paquete. Si tenemos asignado un valor demasiado pequeño para MTU, el rendimiento de NFS se verá muy afectado. Para determinar el valor apropiado de MTU, tiene que:

1. Establecerse como root en el sistema NFS cliente.
2. Ejecutar el comando `tracepath nfsserver/2049`, en el que `nfsserver` es el nombre del host de su servidor NFS. El comando le dirá la ruta a MTU.
3. Compruebe la MTU actual para la interfaz de red que está utilizando, para acceder al servidor NFS. Puede ejecutar el comando `ifconfig` para sacar una lista con información sobre las interfaces de red que está ejecutando.
4. Si observa que el valor de MTU para la interfaz de red, no es el mismo que el que saca el comando `tracepath`, utilice `ifconfig` con la opción `mtu` para establecerlo. Por ejemplo, el comando `ifconfig eth0 mtu 512`, asigna el MTU para la interfaz de red `eth0` en 512 bytes.

## Ejecuta el número óptimo de demonios NFS

Por defecto, ejecuta ocho demonios NFS. Si quiere ver cómo se está utilizando cada hilo nfsd, ejecute el comando `cat /proc/net/rpc/nfsd`. Los últimos diez números de la línea `th` en ese archivo, indican el número de segundos que tenía el manejo del hilo nfsd, como el porcentaje del máximo disponible. Si tiene un número alto en los deciles superiores, debería aumentar el número de instancias nfsd. Para cambiar el número de demonios NFS que se inician cuando se arranca el sistema, haga lo siguiente:

1. Como raíz, pare nfsd utilizando el comando `/etc/rc.d/init.d/nfs stop`, si lo está ejecutando en ese momento.
2. Modifique el script `/etc/rc.d/init.d/nfs`, de modo que `$RPCNFSDCOUNT=8` determine el número adecuado de demonios NFS.
3. Reinicie nfsd utilizando el comando `/etc/rc.d/init.d/nfs start`.

## Controlar el tamaño de la cola de entrada del socket

Por defecto, Linux utiliza una cola de entrada para el socket de 65535 bytes (64KB). Si ejecuta 8 demonios NFS (nfsd) en su sistema, cada demonio obtiene un buffer de 8K para almacenar los datos en la cola de entrada. Debería aumentar el tamaño de la cola hasta, al menos, 256KB, del siguiente modo:

1. Como raíz, pare nfsd utilizando el comando `/etc/rc.d/init.d/nfs stop`, en el caso de que lo esté ejecutando.
2. Modifique el script `/etc/rc.d/init.d/nfs`, de modo que justo antes de iniciar el demonio NFS (nfsd) con la línea `daemon rpc.nfsd $RPCNFSDCOUNT`, se añadan las siguientes líneas:

```
echo 262144 > /proc/sys/net/core/rmem_default  
echo 262144 > /proc/sys/net/core/rmem_max
```

3. Inmediatamente después de la línea `daemon rpc.nfsd $RPCNFSDCOUNT`, añada estas otras líneas:

```
echo 65536 > /proc/sys/net/core/rmem_default  
echo 65536 > /proc/sys/net/core/rmem_max
```

4. Reinicie el demonio NFS utilizando el comando `/etc/rc.d/init.d/nfs start`.

Ahora, cada demonio NFS iniciado por el script `/etc/rc.d/init.d/nfs`, utilizará un espacio de buffer de 32K en la cola de entrada del socket.

## Monitorizar los fragmentos de paquetes

El kernel de Linux controla el número de fragmentos de paquetes UDP sin procesar que se pueden manejar utilizando una serie de más a menos. Cuando el tamaño de los fragmentos de paquetes UDP sin procesar, alcanza el umbral supe-

rior (normalmente 262144 bytes o 256K), el kernel rechaza los fragmentos de paquetes entrantes. En otras palabras, cuando los fragmentos de paquetes UDP alcanzan el umbral máximo, comienza la pérdida de paquetes. La pérdida de fragmentos de paquetes continúa hasta que el tamaño total de fragmentos sin procesar, alcanza el umbral mínimo (normalmente 196608 bytes o 192K).

Como el protocolo NFS utiliza paquetes UDP fragmentados, los umbrales que utiliza Linux influyen enormemente en el rendimiento de NFS. Puede ver el valor actual del tamaño de su máximo umbral ejecutando el comando `cat /proc/sys/net/ipv4/ipfrag_high_thresh`. Del mismo modo, puede ver el valor del umbral inferior ejecutando el comando `/proc/sys/net/ipv4/ipfrag_low_thresh`. Puede cambiar los valores superiores ejecutando `high-number > /proc/sys/net/ipv4/ipfrag_high_thresh`, y del mismo modo, cambiar los inferiores ejecutando `echo low-number > /proc/sys/net/ipv4/ipfrag_low_thresh`.

## Replicar contenido entre servidores Web

Normalmente, el contenido se desarrolla y se sitúa en un solo servidor (maestro) y entonces se distribuye al resto de los servidores que participan en la red Web. En esta sección vamos a ver una herramienta llamada `rdist`, que le permite distribuir contenido desde un sistema Linux a otros sistemas.

## Utilizar `rdist` para distribuir archivos

El programa `rdist` le permite mantener copias idénticas de archivos en varios hosts. Utiliza llamadas a la función `rcmd` o al shell remoto (`rsh`) para acceder a cada computador host.

El modo más sencillo de hacer funcionar a `rdist`, es crear una cuenta común en todas las máquinas implicadas y crear archivos `.rhosts` para cada sistema del servidor Web, de modo que el usuario habitual de dicho servidor, puede ejecutar sesiones `rsh`. Para esto, es necesario crear un usuario llamado `httpd` en los tres sistemas implicados.

En cada uno de estos sistemas del servidor Web, añada un archivo `.rhosts`, en el directorio local del usuario `httpd`. Este archivo contiene el nombre del host del servidor `rdist` en una sola línea.

El archivo `.rhosts` debe pertenecer al usuario `root` y ser de sólo lectura para el resto. Esto le permite a un usuario llamado `httpd` en el servidor `rdist`, ejecutar sesiones shell remotas en cada sistema del servidor Web. Si ha creado una red back-end, tal y como se indicó anteriormente, debería utilizar el nombre del host del servidor `rdist` asociado con la interfaz de red back-end, que le permite mantener el tráfico de archivos generado por `rdist`, en la red back-end; por eso, los paquetes no competirán con su tráfico Web. El siguiente paso es crear un `distfile` para `rdist`. Un `distfile` es un archivo de texto que contiene instrucciones para `rdist`,

sobre cómo realizar las tareas de distribución de archivos. El listado 23.2 muestra uno de estos distfile, el rdist\_distfile.

#### Listado 23.2. rdist\_distfile

```
# Distfile para rdist
#
# Esto se utiliza para distribuir archivos desde ns.domain.com
# a sistemas www[12].domain.com
#
# $Author$ (kabir@nitech.com)
# $Version$
# $Date$
# $Id$

# Realiza una lista de todos los hosts que se tienen que
# actualizar. La lista se crea utilizando entradas user@hostname
# en la que cada entrada se separa con un carácter en blanco.
#
HOSTS = (httpd@www1.domain.com httpd@www2.domain.com)

# Lista de directorios que hay que actualizar.
#
FILES = (/www)

# lista de directorios que hay que excluir del
# proceso de actualización.
EXCLUDE_DIR = (/www/cgi-data/      /www/apache /www/secured)

# A continuación tiene los comandos:
# Instala todos los directorios de la lista FILES para todos
# los hosts de la lista HOSTS excepto para los directorios que
# están en la lista EXCLUDE_DIR
#
${FILES} -> ${HOSTS}
install ;
except ${EXCLUDE_DIR} ;
```

Se trata de un distfile muy sencillo. Define una variable llamada HOSTS que tiene como valores dos entradas: httpd@www1.domain.com y httpd@www2.domain.com. Esto le dice a rdist que utilice la cuenta de usuario httpd en www1.domain.com y en www2.domain.com para conectarse. La siguiente variable, FILES, define los archivos y los directorios para distribuir rdist. Este script supone que el servidor indicado mantiene todos los archivos en el directorio /www o en la partición. Puede cambiar esta ruta o añadir varias rutas (separadas por espacios).

La tercera variable es EXCLUDE\_DIR. Esta variable está dirigida a realizar una lista con todos los archivos y directorios que quiera excluir de la distribución. Los valores que puede ver en el ejemplo son importantes. El primer directorio, /www/cgi-data/, es el directorio de datos CGI en el que todos los scripts CGI

escriben sus datos. Este directorio se exportará al host del servidor Web mediante NFS, de modo que no hay que copiarlo en cada uno de los servidores Web mediante rdist. El directorio /www/apache es en el que makesite escribe el archivo /www/apache/conf/httpd.conf, que tiene que copiarse, porque cada servidor Web tiene su propio archivo de configuración Apache en el directorio local /www/apache. El valor final es /www/secured, el cual es utilizado por el servidor seguro como la raíz de documentos y tiene que copiarse en los servidores Web. El resto de archivos describen un sencillo comando:

```
 ${FILES} -> ${HOSTS}
 install ;
 except ${EXCLUDE_DIR};
```

Este comando toma todos los archivos y directorios que señalan las variables FILES, y los instala en los hosts indicados por la variable HOSTS. También le dice a rdist que excluya los archivos y los directorios especificados por la variable EXCLUDE\_DIR. Para ejecutar rdist (como httpd), utilice el comando siguiente desde la línea de comandos:

```
/usr/bin/rdist -p /usr/sbin/rdistd \
    -oremove,quiet \
    -f /usr/local/rdist/ rdist_distfile
```

La opción -p especifica la localización del programa rdistd que necesita rdist; la opción -o determina que hay que seguir una o más opciones, en este caso, remove y quiet. La opción remove le dice a rdist que elimine cualquier archivo extraño que encuentre en el sistema objetivo, en los directorios objetivo. Esto proporciona un método sencillo para mantener una copia idéntica en el área de pruebas de cada servidor Web. La opción quiet le dice a rdist que se mantenga tan reposado como pueda durante la operación. La opción final, -f, especifica la localización de distfile.

Para reducir los errores humanos en la ejecución de este comando, puede crear un script sh llamado rdistribute.sh, tal y como se muestra en el listado 23.3.

#### Listado 23.3. rdistribute.sh script

```
#!/bin/sh
#
# Este script ejecuta rdist para actualizar servidores Web
# mediante la LAN no routable domain.com. El script se ejecuta
# como cron en un intervalo fijo.
#
# /etc/rc.d/rc.local inicia el script para limpiar los
# tempfiles sobrantes y apagar.
# Este proceso también elimina el
# archivo de registro.
```

```

#
# $Author$ (kabir@evoknow.com)
# $Version$
# $Id$
# $Date$
# $Status
#####
#####

RDIST=/usr/bin/rdist
RDISTD=/usr/sbin/rdistd
DIST_FILE=/usr/local/rdist/rdist_distfile
RDIST_OPTIONS=remove,nochkgroup,nochkmode,nochkowner,quiet
RDIST_LOCK_FILE=/tmp/rdist.lck
RDIST_LOG_FILE=/tmp/rdist.log
TOUCH_BIN=/bin/touch
DATE='date'

# Si el script es llamado como un argumento, entonces
case "$1" in
    boot)
        # Como el argumento es 'boot,' se llama al script en el
        # arranque del sistema, de modo elimine todos los archivos de
        # bloqueo antiguos y los registros.
        echo -n "Cleaning up rdistribute.sh tmp files: "
        rm -f $RDIST_LOCK_FILE
        rm -f $RDIST_LOG_FILE
        echo "complete."
        exit 0;
        ;;

        # Como el argumento es 'restart,' el script
        # tiene que limpiarse como si se acabase de iniciar el
        # sistema.
        restart)
        $0 boot
        ;;

esac

# Si existe el archivo de bloqueo, entonces no hace nada.
if [ -f $RDIST_LOCK_FILE ]; then
    exit 0
fi

# En caso contrario, crea el archivo de bloqueo utilizando
touch
$TOUCH_BIN $RDIST_LOCK_FILE

# Ejecute rdist
$RDIST -p $RDISTD -o$RDIST_OPTIONS -f $DIST_FILE

# Archivo de bloqueo
rm -f $RDIST_LOCK_FILE

```

```
# Escribe la fecha y la hora en el archivo de registro
echo $DATE >> $RDIST_LOG_FILE

# Sale del script
exit 0
```

Este script es lo suficientemente inteligente como para detectar en progreso el proceso `rdistribute.sh` utilizando un archivo de bloqueo, el cual puede decir cuando está listo un `rdistribute.sh` en progreso y continuar. Esto puede tener lugar cuando se actualiza un gran número de archivos en varios servidores. El script también acepta un argumento llamado `boot` que se puede utilizar para limpiar el archivo de bloqueo y el archivo de registro que crea durante el proceso de arranque. Hay que llamar al script desde `/etc/rc.d/rc.local`, del siguiente modo:

```
/usr/local/rdistribute.sh boot
```

Este script se puede programar para ser ejecutado por una entrada de cron en `/etc/crontab`. Por ejemplo, para ejecutar este script a intervalos de 10 minutos, se puede añadir la siguiente entrada cron en `/etc/crontab`:

```
0,10,20,30,40,50 * * * * httpd /usr/local/rdistribute.sh > /
dev/null
```

El demonio cron ejecutará el script como `httpd`.

## Crear un sistema de archivos basado en RAM

Puede crear un pequeño sistema de archivos temporal en RAM para accesos de alta velocidad. El sistema de archivos tiene que ser pequeño porque, por defecto, la máxima cantidad de RAM que puede utilizar ramfs, es la mitad del total de RAM de su sistema. Por eso, si tiene 2GB de RAM, ramfs sólo puede utilizar 1GB. Como aún no he visto sistemas con más de 4GB de RAM, incluso una ramfs de 2GB es realmente pequeña comparado con los grandes sistemas de archivos basados en disco de hoy en día. El ramfs es perfecto para muchos archivos pequeños a los que hay que acceder rápido. Por ejemplo, si utilizo un conjunto de imágenes pequeñas utilizadas en un sitio Web a la que se accede con mucha frecuencia.

## Activar un sistema de archivos basado en RAM

Para utilizar ramfs, debe activar soporte ramfs en el kernel del siguiente modo:

1. Obtener la última fuente del kernel de Linux desde [www.kernel.org](http://www.kernel.org) y extraerlo en el directorio `/usr/src/linux-version` como raíz, donde `version` es la versión actual del kernel. Para el siguiente material, he supuesto que es 2.4.1.

2. Seleccione el submenú **File systems**. Utilizando la barra espaciadora, seleccione **Simple RAM-based file system support**, para incluirlo como un módulo kernel y salga del submenú.
3. Asegúrese de que el resto de las características kernel que utiliza, están seleccionadas también por defecto.
4. Salga del menú principal y guarde la configuración del kernel.
5. Ejecute el comando `make dep` tal y como se indica en el programa menuconfig.
6. A continuación, ejecute `make bzImage` para crear el nuevo kernel. Entonces ejecute `make modules` y `make modules_install` para instalar los módulos nuevos en la localización adecuada.
7. Cambie el directorio a `arch/i386/boot`. Si su arquitectura de hardware es Intel, tiene que reemplazar `i386` y puede que necesite algunas instrucciones más de la documentación sobre el kernel, para compilar e instalar su versión del kernel. Voy a suponer que la mayoría de los sistemas de los lectores están basados en `i386`.
8. Copie `bzImage` en `/boot/vmlinuz-2.4.1` y edite el archivo `/etc/lilo.conf` para incluir una nueva configuración como la siguiente:

```
image=/boot/vmlinuz-2.4.1
      label=linux3
      read-only
      root=/dev/hda1
```

9. Ejecute el comando `/sbin/lilo` para volver a configurar `lilo` y reinicie su sistema. Desde el prompt `lilo` introduzca `linux3` e inicie el nuevo kernel. Si tiene algún problema, debería ser capaz de reiniciar su kernel Linux estándar, que debería estar por defecto.
10. Cuando haya iniciado el kernel nuevo, estará preparado para utilizar `ramfs`. Tiene que crear un directorio llamado `ramdrive` utilizando el comando `mkdir /ramdrive`.
11. Ahora monte el sistema de archivos `ramfs` con el comando `mount -t ramfs none /ramdrive`.

Ahora puede escribir archivos en `/ramdrive` de la forma habitual.

**ADVERTENCIA:** Recuerde que cuando se reinicia el sistema o cuando desmonta el sistema `ramdrive`, todos los datos que se hayan escrito en el directorio `/ramdrive` se perderán. Si tiene datos importantes, no los guarde en `/ramdrive`. Advertencia: si se pierden los datos en `/ramdrive`, no se podrán recuperar. No pierda los datos que necesita.

está utilizando RAM utilizando el comando cat /proc/mounts y buscando una entrada como la siguiente:

```
none /ram ramfs rw 0 0
```

**TRUCO:** Puede especificar opciones, utilizando la opción -o cuando monta el sistema de archivos, del mismo modo que cuando montamos un archivo basado en un disco normal. Por ejemplo, para montar el sistema de archivos ramfs como sólo lectura, puede utilizar la opción -o ro. También puede especificar opciones especiales, como maxsize=n donde n es el número de kilobytes asignados al sistema de archivos en la RAM; maxfiles=n donde n es el número de todos los archivos activados en el sistema de archivos; y maxinodes=n donde n es el número máximo de partes o inodes (el valor por defecto es 0 = sin límites).

## Utilizar el sistema de archivos basado en RAM

Si ejecuta un servidor Web, existen muchos usos para un sistema de archivos basado en RAM. Se pueden mantener en el sistema de archivos ramfs, elementos como imágenes y archivos de su sitio Web que no sean demasiado grandes (no más de unos cuantos KB).

Puede escribir un script shell sencillo, para copiar el contenido desde la localización original en cada reinicio. El listado 23.4 es un sencillo script que lleva esto a cabo.

Listado 23.4. make\_ramfs.sh

```
#!/bin/sh
#
# Script para crear un sistema de archivos ramfs
# on $MOUNTPOINT (which must exists).
#
# Copia archivos de $ORIG_DIR a $MOUNTPOINT
# y cambia el dueño de $MOUTPOINT a
# $USER y $GROUP
#
# Cambia valores para estas variables para adecuarse
# a sus necesidades.

MOUNTPOINT=/ram
ORIG_DIR=/www/commonfiles

USER=httpd
GROUP=httpd
```

```

MOUNTCMD=/bin/mount
CHOWN=/bin/chown
CP=/bin/cp

echo -n "Creating ramfs file system in $MOUNTPOINT ";
$MOUNTCMD -t ramfs none $MOUNTPOINT
echo "done.";

echo -n "Copying $ORIG_DIR to $MOUNTPOINT ... ";
$CP -r $ORIG_DIR $MOUNTPOINT
echo "done.";
echo -n "Changing ownership to $USER:$GROUP for $MOUNTPOINT
...";
$CHOWN -R $USER:$GROUP $MOUNTPOINT
echo "done.";

```

Para utilizar este script en su sistema, haga lo siguiente:

1. Tiene que crear `make_ramfs.sh` en su directorio `/usr/local/scripts`. Tiene que crear el directorio `/usr/local/scripts` si aún no tiene uno.
2. Edite el archivo `/etc/rc.d/rc.local` y adjúntele la siguiente línea:  
`/usr/local/scripts/make_ramfs.sh`
3. Cree un directorio llamado `ram` utilizando el comando `mkdir /ram`. Si mantiene los archivos que quiere cargar en la RAM en una localización distinta de `/www/commonfiles`, entonces modifique el valor de la variable `ORIG_DIR` en el script. Por ejemplo, si sus archivos están en el directorio `/www/mydomain/htdocs/common`, entonces exija que esta variable señale a este directorio.
4. Si ejecuta su servidor Web utilizando un nombre de usuario o un grupo distintos de `httpd`, entonces tiene que cambiar los valores de las variables `USER` y `GROUP`. Por ejemplo, si ejecuta Apache como `nobody` (usuario y grupo), entonces asigne `USER=nobody` y `GROUP=nobody`.
5. Si está utilizando un servidor Web Apache, tiene que crear un alias en su archivo `httpd.conf` del siguiente modo:

```
Alias /commonfiles/ "/ram/commonfiles/"
```

Cada vez que un servidor Web Apache tiene que acceder a `/commonfiles/*`, utilizará ahora la versión en la RAM, que debería ser bastante más rápida que los archivos almacenados en la localización original. Recuerde que la versión basada en la RAM desaparecerá cada vez que reinicie o desmonte el sistema de archivos. Por lo tanto, no actualice nunca nada allí a no ser que copie también los contenidos en un directorio basado en el disco.

**ADVERTENCIA:** Si montó un sistema de archivos ramfs utilizando un comando como `mount -t ramfs none /ram` y copió el contenido en él y más tarde volvió a ejecutar el mismo comando de montaje, la ejecución del mismo comando destruye el contenido y vuelve a montar el ramfs. El archivo `/proc/mounts` muestra varias entradas para el mismo punto de montaje. Lo que da lugar a problemas cuando desmontamos el dispositivo. Si tiene que recuperar la memoria para un uso distinto, tiene que reiniciar. Se espera que todo esto se solucione pronto.

## Crear una red back-end segura

Una red Web típica consiste en un conjunto de servidores Web, servidores proxy, servidores de aplicación, servidores de bases de datos, y este tipo de servidores. En la mayoría de los casos, únicamente los servidores Web y los servidores proxy (en el caso de haberlos) tienen que estar en el lado del cliente, en el front-end de la red Web. La figura 23.3 muestra un ejemplo de una red Web con una red front-end y una red back-end.

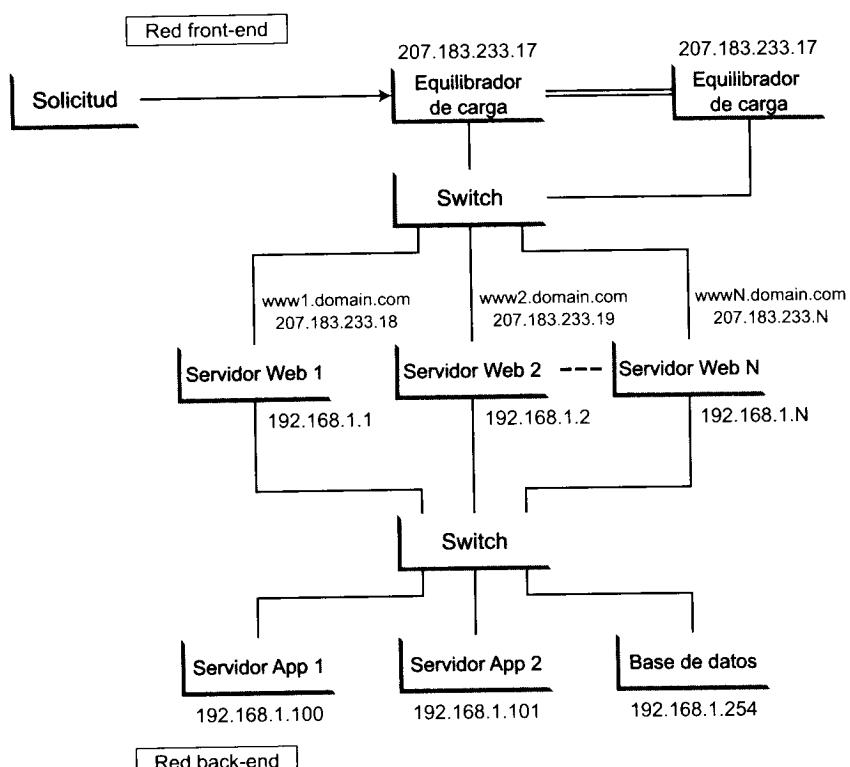


Figura 23.3. Una red Web con red front-end y red back-end

Cada nodo del servidor Web tiene dos interfaces de red. Una interfaz se conecta con la red front-end 207.183.233.0/24 y la otra se conecta con la red back-end 192.168.1.0/24. La red front-end consiste en los servidores Web y el hardware de balance de carga; la red back-end consiste en los servidores Web, servidores de aplicación y servidores de bases de datos.

Cuando llega una solicitud al equilibrador de carga activo (principal), tiene lugar la siguiente cadena de acciones:

- El equilibrador de carga determina el mejor servidor Web para servir solicitudes y pasar la solicitud al servidor Web apropiado.
- El servidor Web decide si la solicitud necesita cualquier recurso adicional. Por ejemplo, si la solicitud es para ejecutar un servidor de aplicación, el servidor Web selecciona el servidor de aplicación apropiado y/o el servidor de bases de datos y realiza las tareas necesarias para completar la solicitud.

No es necesario acceder a los servidores de aplicación y de bases de datos directamente mediante la Web, porque únicamente los servidores Web de la red front-end contactan con ellos. Esto garantiza una buena seguridad y por lo tanto es una buena idea utilizar una dirección de red no ruteable 192.168.x.x para todas las redes back-end adjuntas a la red Web.

Sin embargo, una red Web grande, necesita que los servidores de aplicación y los servidores de bases de datos tengan equilibrio de carga, en caso contrario, la aplicación o la utilización de la base de datos se convierte en el cuello de botella principal en los sitios Web muy ocupados. La figura 23.4 muestra una red Web que utiliza hardware de balance de carga para las redes front-end y back-end.

## Fortificar su red Web

Los piratas informáticos consideran a las redes Web o a cualquier conjunto de nodos que forman un servicio de Internet, como candidatos ideales para ataques. Esto es debido a que si consiguen entrar en una red de computadoras, la red se puede utilizar como plataforma para atacar otros sistemas. En general, una red Web que tenga una mala gestión de seguridad y que esté bien conectada (gran ancho de banda), proporciona un buen recurso para atacar a otras redes. Esto puede causar serios problemas a los administradores del sistema. Por lo tanto, es necesario que lleve a cabo todas las medidas preventivas que pueda para reducir el riesgo de ser atacado.

Remítase al capítulo 18 para obtener los detalles sobre la seguridad de un servidor Web. Esta sección proporciona información adicional relacionada con la seguridad que es adecuada para redes Web, así como para un sistema con un solo servidor.

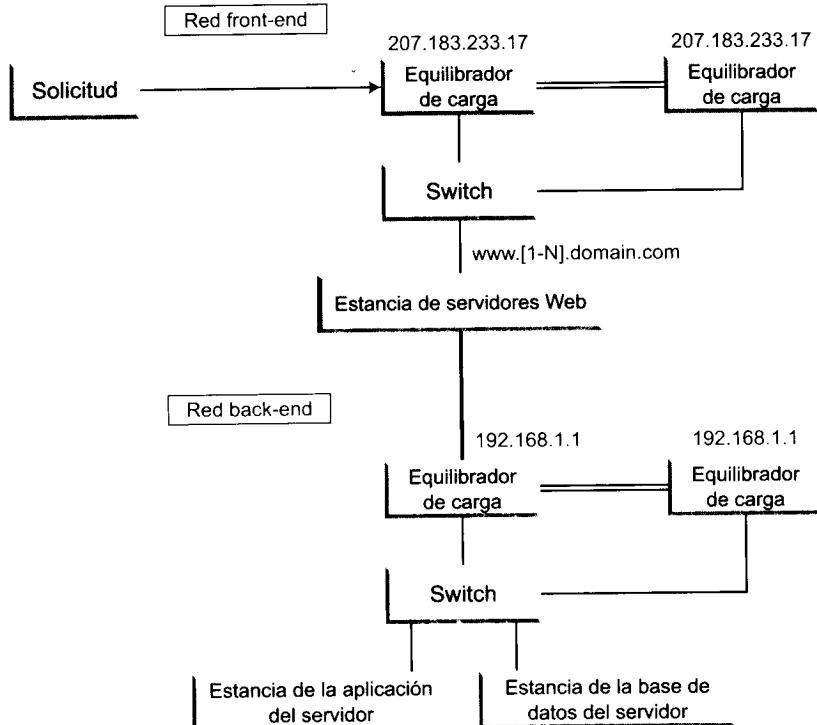


Figura 23.4. Una red Web con equilibradores de carga front-end y back-end

## Utilizar Tripwire para proteger el contenido Web

En un gran movimiento hacia el software de código fuente abierto, Tripwire estrenó Tripwire Open Source de Linux Edition bajo la licencia General Public License (GPL). Hablando en general, Tripwire es un verificador integrado de archivos y de directorios, que crea una base de datos de firmas para todos los archivos y directorios, y las almacena en un solo archivo. Cuando se ejecuta de nuevo, calcula nuevas firmas para los archivos y los directorios actuales y los compara con las firmas originales almacenadas en la base de datos. Si encuentra alguna discrepancia, informa sobre el nombre del archivo o del directorio junto con la información sobre la discrepancia.

A continuación, vamos a ver por qué Tripwire puede ser una fantástica herramienta para ayudarle a determinar qué archivos se han modificado en una intrusión. Por supuesto, tiene que garantizar la seguridad de la base de datos que utiliza la aplicación. Cuando establece un nuevo sistema en el servidor, muchos administradores experimentados, lo hacen en el siguiente orden:

1. Se aseguran de que el sistema nuevo no está adjuntado a ninguna red, con el fin de garantizar que nadie ha instalado un programa Trojano, un virus u otro peligro para la seguridad del sistema.

2. Ejecutan Tripwire para crear una base de datos firmada, de todos los archivos importantes del sistema, incluidos todos los binarios del sistema y los archivos de configuración.
3. Escriben la base de datos en un CD-ROM regrabable. Esto garantiza que nadie puede modificar la base de datos Tripwire para esconder Trojano y proteger al programa de archivos de identidad modificados. Los administradores que tienen que controlar un pequeño número de archivos suelen utilizar un floppy para almacenar la base de datos. Una vez que han escrito la base de datos en el floppy, el disco se protege contra la escritura y, si lo permite la BIOS, el disco duro se configura como un dispositivo de sólo lectura.
4. Establecen un trabajo cron para ejecutar Tripwire periódicamente (diariamente, semanalmente, mensualmente), de modo que la aplicación utiliza la versión del CD-ROM de la base de datos.

## Obtener Tripwire

Red Hat Linux contiene el archivo binario RPM de Tripwire. Sin embargo, siempre puede bajar la versión gratuita (LGPL) de Tripwire de un sitio mirror RPM como <http://fr.rpmfind.net>. Yo bajé el código fuente de Tripwire y los binarios de este sitio, utilizando <http://fr.rpmfind.net/linux/rpm2html/search.php?query=Tripwire>. La fuente RPM que bajé carecía de algunos scripts de instalación, por lo que tuve que bajar de nuevo la fuente desde el sitio de desarrollo de Tripwire Open Source en <http://sourceforge.net/projects/tripwire/>.

El código fuente que bajé se llamaba `tripwire-2.3.0-src.tar.gz`. Podía encontrar la última versión en este sitio en el momento en que este texto fue escrito.

## Compilar Tripwire

Esta sección le muestra cómo compilar, configurar e instalar Tripwire desde el archivo `tripwire-2.3.0-src.tar.gz`. Cuando siga los pasos que se describen a continuación, asegúrese de reemplazar el número de la versión con la versión de Tripwire que haya bajado.

**NOTA:** Si desea instalar Tripwire desde el paquete RPM binario, simplemente ejecute el comando `rpm -ivh tripwire-version.rpm` desde el directorio en el que está localizado Tripwire RPM. Seguirá teniendo que configurar Tripwire ejecutando `twinstall.sh`. Ejecute este script desde el directorio `/etc/tripwire` y salte al paso 7 en la sección siguiente.

Para compilar desde la distribución fuente, realice lo siguiente:

1. Como raíz, extraiga el tar ball utilizando el comando `tar xvzf tripwire-2.3.0-src.tar.gz`. Esto creará un subdirectorio llamado `/usr/src/redhat/SOURCES/tripwire-2.3.0-src`. Cambie su directorio actual al directorio `/usr/src/redhat/SOURCES/tripwire-2.3.0-src/src`.
2. Ejecute el comando `make release` para compilar todos los binarios Tripwire necesarios. Esto le llevará un tiempo. Una vez que tiene compilado Tripwire, tiene que instalar los binarios. Cambie su directorio a `/usr/src/redhat/SOURCES/tripwire-2.3.0-src/install`. Copie los archivos `install.cfg` y `install.sh` en el directorio padre con el comando `cp install.* ...`
3. Antes de ejecutar el script de instalación, tiene que editar el archivo `install.cfg`, que se muestra en el listado 23.5. Por ejemplo, si no está en un editor vi del mundo emacs, tiene que cambiar el campo `TWEDITOR` de este archivo, para que esté dirigido a emacs en lugar de a `/usr/bin/vi`. No le recomiendo cambiar los valores de `CLOBBER`, `TWBIN`, `TWPOLICY`, `TWMAN`, `TWDB`, `TWDOCS`, `TWSITEKEYDIR` o `TWLOCALKEYDIR`. Sin embargo, debería cambiar los valores de `TWLATEPROMPTING`, `TWLOOSEDIRCHK`, `TWMAILNOVIOLATIONS`, `TWEMAILREPORTLEVEL`, `TWREPORTLEVEL`, `TWSYSLOG`, `TWMAILMETHOD` y `TWMAILPROGRAM`.

#### Listado 23.5. `install.cfg`

```
#  
# install.cfg  
#  
# install.cfg para:  
# Tripwire(R) 2.3 Open Source for Linux  
#  
# NOTA: Este es un script shell Bourne que almacena parámetros  
# de instalación para su instalación. El instalador  
# ejecutará este archivo para generar su archivo config  
# y también localizar cualquier necesidad especial para  
# su instalación. Proteja este archivo, porque se puede  
# insertar código no deseado.  
#  
# Esta versión de Tripwire se ha modificado para que sea  
# compatible con el estándar FHS para los sistemas operativos  
# del tipo Unix.  
# Para cambiar el directorio install para cualquier tipo de  
# archivos tripwire, cambie la ruta del modo necesario.  
#  
#=====
```

```
# Si CLOBBER es true, entonces se invalidan los archivos
# existentes. Si CLOBBER es false, entonces no se invalidan los
# archivos existentes.
CLOBBER=false

# Los binarios Tripwire se almacenan TWBIN.
TWBIN="/usr/sbin"

# Los archivos de politicas Tripwire se almacenan en TWPOLICY.
TWPOLICY="/etc/tripwire"

# Las páginas del manual Tripwire se almacenan en TWMAN.
TWMAN="/usr/man"

# Los archivos de la base de datos Tripwire se almacenan en TWDB.
TWDB="/var/lib/tripwire"

# Directorio de documentos Tripwire
TWDOCS="/usr/doc/tripwire"

# Los archivos clave del sitio Tripwire se almacenan en TWSITEKEYDIR.
TWSITEKEYDIR="${TWPOLICY}"

# Los archivos clave locales de Tripwire se almacenan en
# TWLOCALKEYDIR.
TWLOCALKEYDIR="${TWPOLICY}"

# Los archivos de informe de Tripwire se almacenan en TWREPORT.
TWREPORT="${TWDB}/report"

# Asigna el editor de texto por defecto para Tripwire.
TWEDITOR="/bin/vi"

# TWLATEPROMTING controla el momento en el que tripwire pide
# una contraseña.
TWLATEPROMTING=false

# TWLOOSEDIRCHK determina si debería monitorizarse el
# directorio en cuanto a las propiedades que cambian cuando los
# archivos del directorio están monitorizados.
TWLOOSEDIRCHK=false

# TWMAILNOVIOLATIONS determina si Tripwire envía un informe de
# no violación, cuando la verificación de integridad se ejecuta
# con --email-report pero no encuentra ninguna regla violada.
# Esto le permite al administrador saber que hay integridad.
TWMAILNOVIOLATIONS=true

# TWEMAILREPORTLEVEL determina el nivel de detalle (verbosity)
# del informe del e-mail.
TWEMAILREPORTLEVEL=3

# TWREPORTLEVEL determina el nivel de detalle (verbosity) de los
```

```

# informes de impresión.
TWREPORTLEVEL=3

# TWSYSLOG determina si Tripwire registrará eventos en el
# registro del sistema
TWSYSLOG=false

#####
# Opciones de correo: Elige el método
# apropiado y comenta la otra sección
#####

#####
# Opciones SENDMAIL: DEFAULT
#
# Tanto SENDMAIL como SMTP se pueden utilizar para enviar
# informes mediante TWMAILMETHOD.
# Determina qué programa sendmail utilizar.
#####

TWMAILMETHOD=SENDMAIL
TWMAILPROGRAM="/usr/lib/sendmail -oi -t"

#####
# Opciones SMTP
#
# TWSMTPHOST selecciona el host SMTP que hay que utilizar para
# enviar informes.
# SMTPPORT selecciona el puerto SMTP para el programa SMTP mail.
#####

# TWMAILMETHOD=SMTP
# TWSMTPHOST="mail.domain.com"
# TWSMTPPORT=25

#####
# El copyright (C) 1998-2000 Tripwire (R) Security Systems, Inc.
# Tripwire (R) es una marca registrada del Purdue Research
# Foundation y tiene licencia exclusivamente para Tripwire (R)
# Security Systems, Inc.
#####

```

4. Ejecute el comando `./install.sh`. Este comando le introduce en el proceso de instalación. Le pedirán que presione la tecla Enter para aceptar el acuerdo de licencia GPL y para confirmar las localizaciones en las que se copiarán los archivos.
5. Una vez que se han copiado los archivos, le pedirán que introduzca una frase de paso para el sitio. Esta frase de paso se utiliza para encriptar la configuración Tripwire y los archivos de políticas. Introduzca una frase de paso difícil (es decir, una que no sea sencilla de adivinar y que tenga al

menos ocho caracteres), para garantizar que estos archivos no se puedan modificar. A continuación, elija una frase de paso local. Esta frase de paso se utiliza para encriptar la base de datos Tripwire y para archivos de informe. De nuevo, elija una frase de paso difícil.

6. Introduzca la frase de paso y el instalador firma el archivo de configuración utilizando su frase de paso. Se crea una versión de texto claro del archivo de configuración Tripwire en /etc/tripwire/twcfg.txt. La versión binaria del archivo de configuración encriptado, que es lo que utiliza Tripwire, se almacenará en /etc/tripwire/tw.cfg. La versión de texto claro se crea para su inspección. El instalador recomienda que elimine este archivo manualmente una vez que lo ha examinado.
7. Introduzca la frase de paso de modo que el instalador pueda utilizarlo para firmar el archivo de políticas. El instalador crea un archivo de políticas en texto claro, en /etc/tripwire/twpol.txt y la versión encriptada se guarda en /etc/tripwire/tw.pol. Aprenderá a modificar la versión de texto del archivo de políticas más tarde y será capaz de crear la versión encriptada del binario, que es lo que utiliza Tripware.

Esto es lo todo que hay que hacer para instalar el software.

## Configurar la política Tripwire

El archivo de políticas define reglas que utiliza Tripwire para llevar a cabo verificaciones integrales. Cada regla define archivos y directorios que hay que verificar y el tipo de comprobación que hay que realizar. Además, cada regla puede incluir información como el nombre y el nivel de rigor. La sintaxis de una regla es la siguiente:

```
(attribute=value attribute=value ...)  
{  
    /path/to/a/file/or/directory      -> mask;  
}
```

La tabla 23.1 muestra la lista de atributos disponibles y sus significados.

**Tabla 23.1.** Lista de atributos disponibles

Atributo	Significado
rulename=name	Este atributo asocia un nombre con una regla. Realiza informes Tripwire más fáciles de leer y de organizar con dichas reglas.
mailto=e-mailaddress	Cuando se viola una regla, la dirección de correo electrónico para este atributo, recibe un informe.

Atributo	Significado
severity=number	Este atributo le permite asociar un nivel de rigor, es decir, el nivel de importancia, a una regla. Esto convierte los informes Tripwire en informes más fáciles de manejar.
recurse=true   false	Este atributo determina si un directorio se vuelve a procesar automáticamente o no. Si tiene asignado el valor true (o -1), todos los subdirectorios se vuelven a procesar; por otro lado, si tiene asignado el valor false (o 0), los subdirectorios no son atravesados. Cualquier valor numérico en el rango de -1 a 1000000 (excluyendo el -1 y el 0) indica la profundidad de los subdirectorios que se ejecutan. Por ejemplo <code>recurse=3</code> significa que los atraviesan los subdirectorios por encima del nivel de profundidad 3.

Este es un ejemplo de una regla:

```
(RuleName= "OS Utilities", severity=100)
{
    /bin/ls      -> +pinugtsdrbamcCMSH-1;
}
```

la regla definida se llama "OS Utilities"; tiene un rigor de 100, que significa que la violación de esta regla sería considerada como un problema de gravedad; se han verificado las propiedades `+pinugtsdrbamcCMSH-1` de `/bin/ls`. La tabla 23.2 muestra el significado de cada uno de los caracteres de propiedad / máscara.

**Tabla 23.2.** Caracteres propiedad / máscara utilizados en un archivo de políticas Tripwire

Propiedad o máscara	Descripción
a	Temporizador de acceso al archivo o al directorio.
b	Número de bloques asignados al archivo.
c	Temporizador inode.
d	ID del disco en el que reside el inode.
g	Grupo del propietario.
i	Número del inode.
l	El archivo aumenta en tamaño.

Propiedad o máscara	Descripción
m	Temporizador de modificación.
n	Cuenta de referencia de inodes o número de enlaces.
p	Bits de permisos de archivo o directorio.
r	ID del dispositivo señalado por un inode. perteneciente al archivo del dispositivo.
s	Tamaño del archivo.
t	Tipo de archivo.
u	ID del usuario del propietario.
C	Valor CRC-32.
H	Valor Haval.
M	Valor MD5.
S	Valor SHA.
+	Registra y comprueba la propiedad seguida por este carácter.
-	Ignora la propiedad seguida por este carácter.

Otro modo de escribir la regla anterior es:

```
/bin/ls -> +pinugtsdrbamcCMSh-1 (Rulename= "OS Utilities",
severity=100);
```

Sin embargo, es preferible el primer método porque le permite agrupar varios archivos y directorios bajo una sola regla. Por ejemplo,

```
SEC_CRIT = +pinugtsdrbamcCMSh-1;
(Rulename= "OS Utilities", severity=100)
{
    /bin/ls      -> $(SEC_CRIT);
    /bin/login   -> $(SEC_CRIT);
    /bin/ls      -> $(SEC_CRIT);
    /bin/mail    -> $(SEC_CRIT);
    /bin/more    -> $(SEC_CRIT);
    /bin/mt      -> $(SEC_CRIT);
    /bin/mv      -> $(SEC_CRIT);
    /bin/netstat -> $(SEC_CRIT);
}
```

Todas las utilidades de la lista tienen la misma política. Observe la utilización de la variable SEC\_CRIT, que está definida antes de utilizarla en la regla. Esta

variable está asignada a +pinugtsdrbamcCMSh-1 y sustituida en las sentencias de la regla utilizando \$(SEC\_CRIT). Esto le permite definir una sola variable con un conjunto de propiedades que se pueden aplicar a un gran grupo de archivos y/o directorios. Cuando quiera añadir o eliminar propiedades, simplemente cambie el valor de la máscara de la variable; el cambio se refleja en cualquier sitio en el que se utilice la variable. Hay variables integradas que se muestran en la tabla 23.3.

**Tabla 23.3.** Variables integradas para el archivo de políticas Tripwire

Variable	Significado
ReadOnly	+pinugtsdbmCM-rlacSH. Para archivos que permanecen de sólo lectura.
Dynamic	+pinugtd-srlbamccMSh. Para directorios y archivos de usuarios que son dinámicos en cuanto a cambios.
Growing	+pinugtdl-srbamccMSh. Para archivos que crecen en tamaño.
Device	+pugsdr-intlbamccMSh. Para archivos de dispositivos.
IgnoreAll	-pinugtsdrlbamcCMSh. Comprueba si existe el archivo y nada más.
IgnoreNone	+pinugtsdrbamcCMSh-1. Lo contrario de IgnoreAll. Comprueba todas las propiedades.

Cuando se crea una regla, debe considerar los siguientes puntos:

- No tiene que crear varias reglas que se apliquen al mismo archivo o directorio. Por ejemplo:

```
/usr          -> $(ReadOnly);
/usr          -> $(Growing);
```

Tripwire reclamará la política anterior.

- Se respetará la regla más específica. Por ejemplo:

```
/usr          -> $(ReadOnly);
/usr/local/home -> $(Dynamic);
```

Cuando comprueba un archivo /usr/local/home/filename, las propiedades sustituidas por la variable \$(Dynamic) están comprobadas.

Si crea o modifica reglas, tiene que ejecutar el comando /usr/sbin/twadmin --create-polfile /etc/twpol.txt para generar el archi-

vo de políticas encriptado /etc/tripwire/tw.pol. Le pedirán que introduzca la frase de paso del sitio necesaria para marcar (es decir, encriptar) el archivo de políticas.

## Crear la base de datos Tripwire

Antes de iniciar el archivo de la base de datos Tripwire, ha de estar totalmente seguro de que los piratas informáticos no han modificado los archivos en el sistema actual. Esto se debe a que el mejor momento para crear esta base de datos es cuando su sistema no se ha conectado aún a Internet o a cualquier otra red. Cuando esté seguro de que no se han tocado sus archivos, ejecute el comando siguiente:

```
/usr/sbin/tripwire --init
```

Este comando aplica la política de la lista en el archivo /etc/tripwire/tw.pol y crea una base de datos en var/lib/tripwire/k2.intevo.com.

Una vez que ha creado la base de datos, sitúela, si es posible, en un medio de sólo lectura como un CD-ROM o un floppy que esté protegido frente a escritura después de la copia.

## Proteger el propio Tripwire

Los piratas pueden modificar el binario Tripwire (/usr/sbin/tripwire) o el archivo de políticas /etc/tripwire/tw.pol para ocultar pistas de su trabajo. Para prevenirlo, puede ejecutar la utilidad /usr/sbin/siggen para crear un conjunto de firmas para estos archivos. Para generar una firma para el binario /usr/sbin/tripwire, ejecute el comando /usr/sbin/siggen -a /usr/sbin/tripwire.

Verá algo parecido a la siguiente pantalla:

```
-----  
Signatures for file: /usr/sbin/tripwire  
  
CRC32      BmL3O1  
MD5        BrP2IBO3uAzdbRc67CI16i  
SHA        F1IH/HvV3pb+tDhK5we0nKvFUxa  
HAWL       CBLgPptUYq2HurQ+sTa5tV  
-----
```

Puede guardar la firma en un archivo redirigiéndolo a un archivo. Por ejemplo, /usr/sbin/siggen -a /usr/sbin/tripwire > /tmp/sig.txt almacena la firma en el archivo /tmp/sig.txt. También debe imprimir la firma. No olvide generar también una firma para la propia utilidad siggen. Si sospecha que Tripwire no funciona correctamente, ejecute la utilidad siggen en cada uno de estos archivos y compare las firmas. Si alguna de ellas no coincide; debe reemplazarlas con nuevas copias y lanzar una investigación para la discrepancia que ha tenido lugar.

## Ejecutar Tripwire para detectar integridad en el modo interactivo

Para ejecutar en el modo interactivo, ejecute el comando `/usr/sbin/tripwire --check --interactive`. En este modo, se genera un archivo de informe en su editor preferido. En el listado 26.3 se muestra una parte de un informe Tripwire generado por este comando.

### Listado 23.6. Tripwire report

```
Tripwire(R) 2.3.0 Integrity Check Report
```

```
Report generated by: root
Report created on: Fri Dec 22 02:31:25 2000
Database last updated on: Fri Dec 22 02:13:44 2000
```

```
=====
Report Summary:
=====
```

```
Host name: k2.intevo.com
Host IP address: 172.20.15.1
Host ID: None
Policy file used: /etc/tripwire/tw.pol
Configuration file used: /etc/tripwire/tw.cfg
Database file used: /var/lib/tripwire/k2.intevo.com.twd
Command line used: /usr/sbin/tripwire --check --
interactive
```

```
=====
Rule Summary:
-----
```

```
-----  
Section: Unix File System  
-----
```

Rule Name	Severity	Level	Added	Removed	Modified
Invariant Directories	66		0	0	0
Temporary directories	33		0	0	0
* Tripwire Data Files	100		0	0	1
Critical devices	100		0	0	0
User binaries	66		0	0	0
Tripwire Binaries	100		0	0	0
* Critical configuration files	100		0	0	1
Libraries	66		0	0	0
Shell Binaries	100		0	0	0
File System and Disk Administraton Programs	100		0	0	0
Kernel Administration Programs	100		0	0	0

Networking Programs	100	0	0	0
System Administration Programs	100	0	0	0
Hardware and Device Control Programs	100	0	0	0
System Information Programs	100	0	0	0
Application Information Programs	100	0	0	0
Shell Related Programs	100	0	0	0
Critical Utility Sym-Links	100	0	0	0
Critical system boot files	100	0	0	0
System boot changes	100	0	0	0
OS executables and libraries	100	0	0	0
Security Control	100	0	0	0
Login Scripts	100	0	0	0
Operating System Utilities	100	0	0	0
Root config files	100	0	0	0

Total objects scanned: 14862

Total violations found: 2

Hay dos violaciones que están marcadas con el signo \* a la izquierda de las líneas. La primera violación tiene lugar para la regla Tripwire Data Files. El informe indica también que hay otra violación para la regla Critical configuration files. En ambos casos, se ha modificado un archivo que no tendría que haberse modificado. La sección Object Summary del informe, muestra las líneas siguientes:

```
=====
Object Summary:
=====

-----
# Sección: Unix File System
-----

-----
Rule Name: Tripwire Data Files (/etc/tripwire/tw.pol)
Severity Level: 100
-----

Remove the "x" from the adjacent box to prevent updating the
database
with the new values for this object.

Modified:
[x] "/etc/tripwire/tw.pol"

-----
Rule Name: Critical configuration files (/etc/cron.daily)
Severity Level: 100
-----
```

Remove the "x" from the adjacent box to prevent updating the database with the new values for this object.

Modified:  
[x] "/etc/cron.daily"

Como puede ver, Tripwire muestra exactamente los archivos que se han modificado y qué reglas siguen estos archivos. Si las modificaciones son correctas, la marca 'x' (significa sección) se puede colocar en las secciones apropiadas del informe. Tripwire actualizará la base de datos. Por ejemplo, si colocó las marcas x en ambos archivos, la próxima vez que se ejecute el verificador de integridad, no encontrará estas violaciones porque la base de datos Tripwire fue actualizada para informar sobre estos archivos modificados. Sin embargo, si no se esperaba alguna de estas modificaciones y resulta sospechosa, Tripwire realizará su trabajo.

**TRUCO:** Si quiere ver un informe desde el directorio /var/lib/tripwire/report, puede ejecutar el comando /usr/sbin/twprint -m r --twrfile reportfilename en cualquier momento.

## Ejecutar Tripwire para detectar integridad de forma automática

También puede ejecutar Tripwire como un trabajo de un cron, creando un pequeño script como el que se muestra en el listado 23-6.

Listado 23.6. /etc/cron.daily/tripwire-check

```
#!/bin/sh
HOST_NAME='uname -n'
if [ ! -e /var/lib/tripwire/${HOST_NAME}.twd ] ; then
    echo "***      Error: Tripwire database for ${HOST_NAME} not
found.      ***"
    echo "*** Run "/etc/tripwire/twinstall.sh" and/or "tripwire
--init". ***"
else
    test -f /etc/tripwire/tw.cfg && /usr/sbin/tripwire --check
fi
```

Este script comprueba si existe la base de datos Tripwire; si existe, el script busca el archivo de configuración y cuando lo encuentra, se ejecuta el comando /usr/sbin/tripwire en un modo no interactivo. Esto da lugar a un informe y, si ha configurado una o más reglas utilizando el atributo emailto, los correos electrónicos se enviarán a la persona adecuada.

## Actualizar la base de datos Tripwire

Tiene que actualizar la base de datos Tripwire cada vez que tenga lugar un cambio en los sistemas de archivos que generarán una falsa advertencia de ausencia de la base de datos que se acaba de actualizar. Por ejemplo, si modifica un archivo de configuración, o elimina un programa que Tripwire está "vigilando", Tripwire generará un informe de violación. Por lo tanto, cada vez que cambie algo de forma intencionada, tiene que actualizar la base de datos, bien reiniciando la base de datos con el comando `/usr/sbin/tripwire -init`, o bien utilizando el comando `/usr/sbin/tripwire --update` para, simplemente, actualizar la base de datos. El método de actualización debería ahorrarle algo de tiempo porque no tiene que volver a crear la base de datos completa.

Del mismo modo, cuando cambia el archivo de políticas Tripwire `/etc/tripwire/twpol.txt`, tiene que actualizar la base de datos. De nuevo, en lugar de reiniciar la base de datos completa utilizando la opción `--init`, puede instruir al programa para que aplique los cambios en la política y para que actualice la base de datos con el comando `/usr/sbin/tripwire --update-policy /etc/tripwire/twpol.txt`.

Una vez que ha creado una base de datos tripwire, tiene que actualizarla cada vez que actualice su archivo de políticas. En lugar de reiniciar la base de datos cada vez que cambie (o experimente) con su archivo de políticas, puede ejecutar el comando `tripwire --update-policy /etc/tripwire/twpol.txt` para actualizar la base de datos. Esto le ahorrará una cantidad de tiempo importante.

## Obtener un informe tripwire por correo electrónico

Si utiliza el atributo `mailto` puede recibir informes de violación de las reglas (o incluso de no-violación de las reglas) de Tripwire. Esto resulta de especial utilidad si está ejecutando verificaciones Tripwire como un trabajo de cron (remítase a la sección "Ejecutar Tripwire para detectar integridad de forma automática"). Antes de poder obtener un correo electrónico de Tripwire, debe configurar las opciones de correo electrónico en el archivo `/etc/tripwire/twcfg.txt` y volver a construir el archivo de configuración con el comando `/usr/sbin/twadmin --create-cfgfile /etc/tripwire/twcfg.txt`. Las opciones que controlan el correo electrónico se explican en la tabla 23.4.

**Tabla 23.4.** Opciones de correo electrónico para el archivo de configuración de Tripwire

Atributo	Significado
<code>MAILMETHOD = SMTP   SENDMAIL</code>	Predefinido: <code>MAILMETHOD = SENDMAIL</code> . Este atributo se utiliza para determinar el método de envío de correos electrónicos que utiliza Tripwire.

Atributo	Significado
	<p>El valor por defecto le permite a Tripwire utilizar el demonio Sendmail, que debe especificarse utilizando el atributo MAILPROGRAM que se discute más tarde. Los demonios mail alternativos más conocidos de Sendmail, son qmail y postoffice, funcionan mucho mejor que Sendmail, puede seguir asignándolo a SENDMAIL y especificar la ruta del demonio alternativo utilizando el MAILPROGRAM.</p> <p>Sin embargo, si no ejecuta un demonio Sendmail, o del tipo Sendmail, en la máquina en la que está ejecutando el programa Tripwire, puede asignar este atributo a SMTP y especificar el número de atributos MTPHOST y SMTPPORT. Suponiendo que SMTPHOST le permite a su sistema transmitir mensajes, Tripwire conectará con el host mediante el puerto SMTP y enviará mensajes, los cuales enviará el host más tarde al destino apropiado.</p>
<b>SMTPHOST = hostname   IP Address</b>	<p>Predefinido: ninguno</p> <p>Este atributo le permite especificar el nombre del host de un servidor de correo. Utilícelo únicamente si no tiene capacidades de correo en el mismo sistema en el que se ejecuta Tripwire. Puede ver la dirección IP del servidor de correo o el nombre del host, utilizando el comando nslookup -q=mx yourdomain.</p>
<b>SMTPPORT = port number</b>	<p>Predefinido: ninguno</p> <p>Este atributo especifica el número de puerto TCP del servidor de correo remoto. Normalmente, tiene un valor de 25. Sólo lo necesita si asigna MAILMETHOD a SMTP.</p>
<b>MAILPROGRAM = /path/to/mail/program</b>	<p>Predefinido: MAILPROGRAM = /usr/sbin/sendmail -oi -t</p> <p>Este atributo especifica la ruta de cualquier argumento que necesite suministrar para ejecutarlo. Este atributo sólo tiene sentido si está utilizando MAILMETHOD = SENDMAIL.</p>
<b>EMAILREPORTLEVEL = 0 – 4</b>	<p>Predefinido: EMAILREPORTLEVEL = 3</p> <p>Este atributo especifica el nivel de información que se distribuye mediante correo electrónico. Deje el valor por defecto.</p>
<b>MAILNOVIOLATIONS = true   false</b>	<p>Predefinido: MAILNOVIOLATIONS = true</p> <p>Si no quiere recibir un correo electrónico cuando no se encuentra violación, asignele el valor false.</p>

Para comprobar sus opciones de correo electrónico, puede ejecutar Tripwire utilizando el comando `/usr/sbin/tripwire -m t -email your@emailaddr`. No olvide cambiar el `you@emailaddr` con su propia dirección de correo electrónico.

## Asegurar Apache utilizando el Intrusion Detection System (LIDS) de Linux

*Root es la fuente de todos sus males.* La frase anterior probablemente sólo tenga sentido para los administradores de los sistemas Unix/Linux. Una vez que se confirma un acceso root, el control de daños parece imposible o a la merced del intruso.

En un sistema Linux vanilla, existen varios subsistemas desprotegidos. El sistema de archivos suele quedar totalmente abierto. Hay archivos importantes, como `/bin/login`, en el sistema, que los piratas suelen explotar porque están desprotegidos. Si un pirata entra, puede descargar una versión modificada del programa de registro como `/bin/login` para permitirse el acceso libre al sistema en el futuro. Pero el problema es que estos archivos (es decir, los programas) como los archivos `/bin/login` no tienen que cambiar frecuentemente (o no cambian en absoluto); por lo tanto, no deben dejarse sin protección. Al igual que es sistema de archivos, los procesos en ejecución también están desprotegidos. Muchos procesos se ejecutan con privilegios de root, lo que significa que cuando están aprovechados utilizando trucos como un desagüe buffer, el intruso obtiene acceso de root completo al sistema.

Reducir el poder del usuario root incrementa la seguridad del sistema. LIDS. Realmente hace otras cuantas cosas más. Implementa un modelo de seguridad de muy bajo nivel en el kernel para proporcionar protección, detecta incidentes y tiene capacidad de respuesta a incidentes. Por ejemplo, LIDS puede:

- Proteger archivos y directorios importantes de accesos no autorizados en su disco duro independientemente del sistema de archivos local en el que resida. Los archivos y directorios elegidos se pueden proteger de modificaciones por parte del usuario root, lo que significa que un acceso root no autorizado no tiene por qué convertir al intruso en un grave peligro.
- Proteger procesos importantes para que nadie pueda finalizarlos, incluyendo al usuario root. De nuevo, esto reduce las capacidades del usuario root.
- Previene operaciones I/O de programas no autorizados. También protege el registro de arranque maestros (MBR) del disco duro.

LIDS puede detectar cuándo alguien analiza su sistema utilizando escáneres de puertos y puede comunicarle al administrador del sistema, mediante un correo electrónico, que se está produciendo un análisis. LIDS también puede mandar una notificación al administrador del sistema, cada vez que nota cualquier violación

de las reglas impuestas. Cuando alguien viola dicha regla, LIDS puede registrar mensajes detallados sobre las violaciones en archivos de registro protegidos por LIDS, archivos de registro de pruebas. De hecho, LIDS no sólo puede registrar y enviar un correo electrónico sobre las violaciones detectadas, puede apagar inmediatamente la sesión de un usuario.

LIDS es un parche del kernel y un conjunto de herramientas administrativas que incrementan la seguridad del kernel del sistema operativo Linux. Como en el modelo de seguridad de LIDS, el sujeto, el objeto y el tipo de acceso están en el kernel, se denomina monitor de referencia. El sitio Web del proyecto LIDS es [www.lids.org/about.html](http://www.lids.org/about.html).

LIDS le permite al sistema Linux ejecutar un kernel personalizado y debe obtener la última fuente del kernel de un sitio de confianza como [www.kernel.org](http://www.kernel.org). Una vez que ha bajado y extraído el kernel en `/usr/src/linux`, baje el parche LIDS para el kernel específico que quiera utilizar. Por ejemplo, si está utilizando el kernel 2.4.1, asegúrese de bajar el parche LIDS del sitio Web del proyecto LIDS. Normalmente el parche LIDS y el paquete de herramientas administrativas se denominan `lids-x.x.x.y.y.y.tar.gz` donde `x.x.x` representa el número de versión de LIDS y `y.y.y` representa la versión del kernel (por ejemplo, `lids-1.0.5-2.4.1`). En las siguientes instrucciones voy a utilizar LIDS 1.0.5 para el kernel 2.4.1. Asegúrese de cambiar los números de versión como sea necesario. Extraiga la distribución fuente de LIDS en el directorio `/usr/local/src` con el comando `tar xvzf lids-1.0.5-2.4.1.tar.gz` desde el directorio `/usr/local/src`. Ahora puede parchear el kernel.

**NOTA:** Asegúrese de que `/usr/src/linux` está dirigido a la última distribución fuente del kernel que ha bajado. Simplemente ejecute `ls -l /usr/src/linux` para ver a qué directorio apuntan los enlaces simbólicos. Si apuntan a una fuente antigua del kernel, elimine el enlace utilizando `rm -f /usr/src/linux` y vuelva a enlazarlo utilizando `ln -s /usr/src/linux-version /usr/src/linux` donde `version` es la versión del kernel que ha bajado. Por ejemplo, `ln -s /usr/src/linux-2.4.1 /usr/src/linux` enlaza la última fuente del kernel 2.4.1 a `/usr/src/linux`.

## Parchear, compilar e instalar el kernel con LIDS

Para compilar LIDS para su sistema, tiene que volver a compilar el kernel de Linux después de aplicar el parche LIDS. Este proceso se describe a continuación:

1. Como root, extraiga el paquete de parches LIDS en un directorio adecuado. Normalmente guardo el código en el directorio `/usr/local/src`.

Para estos pasos, voy a suponer que ha hecho lo mismo; si no utiliza el mismo directorio, realice las modificaciones apropiadas. Desde el directorio /usr/local/src, ejecute el comando tar xvzf lids-1.0.5-2.4.1.tar.gz. Esto creará un nuevo subdirectorio llamado lids-1.0.5-2.4.1.

2. Cambie el directorio /usr/src/linux y ejecute el comando patch -p.< /usr/local/src/lids-1.0.5-2.4.1.patch para parchear la distribución fuente del kernel.
3. Desde el directorio /usr/src/linux ejecute el comando make menuconfig para iniciar el programa de configuración del kernel basado en menú. También puede utilizar los comandos make config o make xconfig para configurar el kernel, pero prefiero la primera opción y voy a suponer que es la que va a utilizar.
4. Desde el menú principal seleccione el submenú Code maturity level options y elija la opción Prompt for development and/or incomplete code/drivers presionando la barra espaciadora. Salga de este menú.
5. Diríjase al submenú General setup, seleccione Sysctl support y salga del submenú.
6. Desde el menú principal, seleccione el submenú Linux Intrusion Detection System, que únicamente aparece si ha completado los pasos 4 y 5. Este submenú debería aparecer abajo del menú principal de modo que tendrá que utilizar la barra de scroll.
7. Desde el submenú LIDS seleccione la opción Linux Intrusion Detection System support (EXPERIMENTAL) (NEW). Verá una lista de opciones como la que se muestra a continuación.

```
(1024) Maximum protected objects to manage (NEW)
(1024) Maximum ACL subjects to manage (NEW)
(1024) Maximum ACL objects to manage (NEW)
(1024) Maximum protected processes (NEW)
[ ] Hang up console when raising a security alert (NEW)
[ ] Security alert when executing unprotected programs
before sealing LIDS (NEW)
[ ] Try not to flood logs (NEW)
[ ] Allow switching LIDS protections (NEW)
[ ] Port Scanner Detector in kernel (NEW)
[ ] Send security alerts through network (NEW)
[ ] LIDS Debug (NEW)
```

8. Los límites por defecto para los objetos manejados, los objetos protegidos, los sujetos / objetos ACL y los procesos protegidos deberían ser adecuados para la mayoría de los sistemas. Por eso debe dejarlos tal y como están.

9. Si quiere que LIDS desconecte la consola cuando un usuario viole alguna regla de seguridad, entonces seleccione la opción Hang up console when raising a security alert.
10. LIDS está activado durante el proceso de arranque, por lo que es probable que se estén ejecutando otros programas antes del arranque. Si quiere emitir una advertencia de seguridad cuando se ejecuta un programa antes de activar la protección LIDS, seleccione la opción Security alert when execing unprotected programs before sealing LIDS. Cuando selecciona esta opción, tendrá la posibilidad de desactivar la ejecución de programas desprotegidos utilizando la opción Do not execute unprotected programs before sealing LIDS. No recomiendo que desactive completamente los programas desprotegidos durante el inicio, a no ser que esté absolutamente seguro de que todo (es decir, todas las utilidades, demonios, y similares) lo que quiere ejecutar durante el arranque, está protegido y no parará el proceso normal de arranque.

11. Active la opción Try not to flood logs (NEW) y deje el valor por defecto de 60 segundos de retardo entre el registro de dos entradas idénticas, para conservar el estado y el tamaño del archivo de registro.
12. (Opcional.) Seleccione la opción Allow switching LIDS protections si quiere permitir cambios en la protección LIDS. Si lo hace, puede personalizarlo seleccionando el valor para Number of attempts to submit password, o Time to wait after a fail (seconds), o Allow remote users to switch LIDS protections, o Allow any program to switch LIDS protections, o Allow reloading config. file. Mis preferencias son las siguientes.

```
[*] Allow switching LIDS protections (NEW)
(3) Number of attempts to submit password (NEW)
(3) Time to wait after a fail (seconds) (NEW)
[*] Allow remote users to switch LIDS protections (NEW)
[ ] Allow any program to switch LIDS protections (NEW)
[*] Allow reloading config. file (NEW)
```

13. Seleccione la opción Port Scanner Detector in kernel de modo que pueda detectar si los intrusos potenciales están escaneando puertos y la opción Send security alerts through network. Deje los valores por defecto para la segunda opción.
14. Guarde la configuración del kernel y ejecute los siguientes comandos para compilar el kernel nuevo y sus módulos (si tiene).

```
make depend
make bzImage
```

```
make modules  
make modules_install
```

**ADVERTENCIA:** Si no está compilando una versión más moderna del kernel de la que se está ejecutando en el sistema, debería hacer un backup del directorio /bin/modules/current-version, donde current-version es la versión actual del kernel. Por ejemplo, si está compilando la versión 2.4.1 y está ejecutando la versión 2.4.1, entonces debería ejecutar el comando cp -r /lib/modules/2.4.1 ./lib/modules/2.4.1.bak para realizar un backup de los módulos actuales. En el caso de un problema con el nuevo kernel, puede eliminar los módulos estropeados del kernel y volver a nombrar este directorio con su nombre original.

15. Ahora copie la imagen creada del kernel /usr/src/linux/arch/i386/boot/bzImage en /boot/vmlinuz-lids-1.0.5-2.4.1 utilizando el comando cp /usr/src/linux/arch/i386/boot/bzImage /boot/vmlinuz-lids-1.0.5-2.4.1.

16. Añada el archivo /etc/lilo.conf:

```
image=/boot/vmlinuz-lids-1.0.5-2.4.1  
label=lids  
read-only  
root=/dev/hda1
```

Si /dev/hda1 no es el dispositivo root, asegúrese de cambiarlo del modo apropiado.

17. Ejecute /sbin/lilo para volver a configurar el lilo.

La parte del kernel de la configuración está completa y se puede realizar la configuración de LIDS.

## Compilar, instalar y configurar LIDS

Para compilar e instalar el programa de administración de LIDS, lidsadm, siga los pasos siguientes:

1. Suponiendo que tenga instalada la fuente LIDS en el directorio /usr/local/src, cambie a /usr/local/src/lids-1.0.5-2.4.1/lidsadm-1.0.5.
2. Ejecute los comandos make; make install para instalar el programa lidsadm en /sbin y para crear los archivos de configuración necesarios (lids.cap, lids.conf, lids.net, lids.pw) en /etc/lids.
3. Ejecute el comando /sbin/lidsadm -P e introduzca una contraseña para el sistema LIDS. Esta contraseña se almacena en el archivo /etc/lids/lids.pw en el formato encriptado RipeMD-160.

- Ejecute el comando `/sbin/lidsadm -U` para actualizar los números inode/dev.
- Configure el archivo `/etc/lids/lids.net`. A continuación tiene una versión simplificada del archivo `/etc/lids/lids.net`:

```
MAIL_SWITCH= 1
MAIL_RELAY=127.0.0.1:25
MAIL_SOURCE=lids.sinocluster.com
MAIL_FROM= LIDS_ALERT@lids.sinocluster.com
MAIL_TO= root@localhost
MAIL SUBJECT= LIDS Alert
```

- La opción `MAIL_SWITCH` puede ser 1 o 0, donde 1 activa la función de alerta de correo electrónico y 0 la desactiva. Deje el valor por defecto.
  - La opción `MAIL_RELAY` debería determinar la dirección IP del servidor de correo que LIDS debería utilizar para enviar el mensaje de alerta. Si ejecuta el servidor de correo en la misma máquina en la que está configurando LIDS, deje el valor por defecto. El número de puerto 25, es el puerto por defecto SMTP y debe mantenerse, a no ser que esté ejecutando su servidor de correo en un puerto distinto.
  - La opción `MAIL_SOURCE` debería estar asignada al nombre del host de la máquina que se está configurando. Cambie el valor por defecto al nombre del host de su sistema.
  - La opción `MAIL_FROM` debería estar asignada a una dirección que le diga desde qué sistema viene la advertencia. El valor por defecto debería cambiarse para reflejar el nombre del host de su sistema. No tiene que crear una verdadera cuenta de correo para que sea útil.
  - La opción `MAIL_TO` debería estar asignada a la dirección de correo electrónico del administrador del sistema que se está configurando. Como la dirección `root`, `root@localhost`, es, por defecto, la cuenta del administrador la puede dejar como está.
  - La opción `MAIL SUBJECT`, es el asunto del correo, y se cambiará según sus necesidades.
- Para ver qué es lo que está protegido por defecto, ejecute el comando `/sbin/lidsadm -L`, que debería mostrar una salida parecida a la siguiente:

LIST

Subject	ACCESS TYPE	Object
Any File	READ	/sbin
Any File	READ	/bin

Any File	READ	/boot
Any File	READ	/lib
Any File	READ	/usr
Any File	DENY	/etc/shadow
/bin/login	READ	/etc/shadow
/bin/su	READ	/etc/shadow
Any File	APPEND	/var/log
Any File	WRITE	/var/log/wtmp
/sbin/fsck.ext2	WRITE	/etc/mtab
Any File	WRITE	/etc/mtab
Any File	WRITE	/etc
/usr/sbin/sendmail	WRITE	/var/log/sendmail.st
/bin/login	WRITE	/var/log/lastlog
/bin/cat	READ	/home/xhg
Any File	DENY	/home/httpd
/usr/sbin/httpd	READ	/home/httpd
Any File	DENY	/etc/httpd/conf
/usr/sbin/httpd	READ	/etc/httpd/conf
/usr/sbin/sendmail	WRITE	/var/log/sendmail.st
/usr/X11R6/bin/XF86_SVGA		
/usr/sbin/in.ftpd		
/usr/sbin/httpd	NO_INHERIT	RAWIO
	READ	/etc/shadow
		HIDDEN

Como probablemente no tenga /home/xhg (el directorio local del autor de LIDS), puede eliminar la configuración para él, utilizando el comando /sbin/lidsadm -D -s /bin/cat -o /home/xhg. Puede dejar todo lo demás tal y como está porque puede cambiarlo más tarde en función de sus necesidades.

7. Añada la siguiente línea al archivo /etc/rc.d/rc.local para cerrar el kernel durante el final del ciclo de arranque:  
`/sbin/lidsadm -I`
8. Reinicie el sistema y elija el kernel LIDS activado, introduciendo lids en el prompt lilo. Cuando el sistema se inicie y ejecute el comando /sbin/lidsadm -I desde el script /etc/rc.d/rc.local, cerrará el kernel y el sistema estará protegido por LIDS.

## Administrar LIDS

Excepto para el archivo /etc/lids/lids.net, debe utilizar el programa /sbin/lidsadm para modificar los archivos de configuración LIDS /etc/lids/lids.conf, /etc/lids/lids.pw y /etc/lids/lids.cap.

El archivo /etc/lids/lids.conf almacena la información Access Control List (ACL). El archivo /etc/lids/lids.cap contiene todas las reglas de capacidad para el sistema. Puede configurar qué capacidad quiere activar o desactivar en el sistema, editando este archivo con el comando /sbin/lidsadm. Puede colocar un signo + enfrente del nombre de la capacidad para activar el sistema, o un signo - para desactivar la capacidad. El archivo /etc/lids/lids.net configura el sistema de correo necesario para enviar correos electró-

nicos de alerta. Puede utilizar un editor normal como vi, emacs o pico, para editar este archivo.

Si tiene que parar LIDS para realizar tareas de administración, entonces debería utilizar el comando `/sbin/lidsadm -S -- -LIDS` o el comando `/sbin/lidsadm -S -- -LIDS_GLOBAL`. También tendrá que proporcionar la contraseña LIDS para apagar LIDS, después de realizar cualquier cambio en el archivo de configuración LIDS con el comando `lidsadm`, vuelva a cargar la configuración actualizada en el kernel ejecutando el comando `/sbin/lidsadm -S -- + RELOAD_CONF`.

Para añadir un nuevo ACL al archivo `/etc/lids/lids.conf`, utilice el comando `/sbin/lidsadm` del siguiente modo:

```
/sbin/lidsadm -A [-s subject] [-t | -d | -i] -o object -j TARGET
```

Las opciones para este comando se explican en la lista siguiente:

- La opción `-A` le dice al programa `/sbin/lidsadm` que añada un nuevo ACL.
- La opción `-s subject` especifica un sujeto para el ACL. Un sujeto puede ser cualquier programa como `/bin/cat`. Cuando no especifica un sujeto, el ACL se aplicará a todo.
- Las opciones `-t`, `-d`, `-i` no se suelen necesitar.
- La opción `-o object` se utiliza para especificar el nombre del objeto, que puede ser un archivo, un directorio o una capacidad. Cada ACL necesita un objeto determinado.
- La opción `-j TARGET` especifica el objetivo de ACL. Cuando el nuevo ACL tiene un archivo o un directorio como objeto, el objetivo puede ser `ob` READ, WRITE, APPEND, DENY y IGNORE. Si el objeto es una capacidad Linux, el objetivo puede ser únicamente INHERIT o NO\_INHERIT, los cuales definen si los hijos de los objetos pueden tener o no la misma capacidad.

## Proteger archivos y directorios

Puede utilizar `lidsadm` para proteger archivos y directorios importantes. Puede hacer un archivo o un directorio de sólo lectura, controlar el acceso a la lectura, controlar el acceso a archivos en el modo append (permite el acceso a archivos que se encuentran en un directorio distinto al directorio actual), y rechazar el acceso. LIDS proporciona el siguiente tipo de protección para un archivo o un directorio.

- READ: convierte el archivo o el directorio en sólo lectura
- WRITE: permite modificaciones en el archivo o en el directorio
- IGNORE : ignora la protección asignada a un archivo o un directorio

- APPEND: permite añadir a un archivo
- DENY: rechaza todos los accesos al archivo o al directorio

## Convertir el archivo o el directorio en sólo lectura

Para hacer que el archivo /path/filename sea sólo de lectura, ejecute:

```
/sbin/lids -A -o /path/filename -j READ
```

Para hacer que el directorio /mypath sea sólo de lectura, ejecute:

```
/sbin/lids -A -o /mypath -j READ
```

Observe que como no especificamos un sujeto en ninguno de los comandos anteriores, el ACL se aplica a todos los programas. Por lo tanto, ningún programa puede escribir en el archivo o en el directorio mencionado. Si especifica un sujeto, entonces el comando sólo se aplicará a dicho archivo o directorio.

## Rechazar el acceso a un archivo o a un directorio

Para denegar el acceso al archivo /etc/shadow, ejecute:

```
/sbin/lids -A -o /etc/shadow -j DENY
```

Cuando se ejecuta este comando y se vuelve a cargar la configuración de LIDS, puede ejecutar comandos del tipo ls -l /etc/shadow y cat /etc/shadow para ver si puede acceder al archivo. Ninguno de estos programas verá el archivo porque implícitamente especificó que el sujeto fuesen todos los programas del sistema. Sin embargo, si tiene que permitir a un programa como /bin/login el acceso al archivo /etc/shadow, puede darle acceso de lectura creando un nuevo ACL como:

```
/sbin/lids -A -s /bin/login -o /etc/shadow -j READ
```

## Permitir acceso en el modo append

Normalmente, los programas sólo necesitan acceso append a registros críticos del sistema como /var/log/messages o /var/log/secure. Puede activar el modo append para estos dos archivos con los comandos siguientes:

```
/sbin/lids -A -o /var/log/messages -j APPEND  
/sbin/lids -A -o /var/log/secure -j APPEND
```

## Permitir acceso de sólo escritura

Para permitirle a un programa llamado /usr/local/apache/bin/httpd escribir en un directorio protegido llamado /home/httpd, ejecute los siguientes comandos:

```
/sbin/lids -A -o /home/httpd -j DENY  
/sbin/lids -A -s /usr/local/apache/bin/httpd -o /home/httpd -j  
READ
```

## **Eliminar un ACL**

Para eliminar todas las reglas ACL, ejecute el comando `/sbin/lidsadm -Z`. Para eliminar una regla ACL concreta, simplemente especifique el sujeto (si hay) y/o el objeto de la ACL. Por ejemplo, si ejecuta el comando `/sbin/lidsadm -D -o /bin`, todas las reglas ACL con `/bin` como objeto, son eliminadas. Sin embargo, si ejecuta el comando `/sbin/lidsadm -D -s /bin/login -o /bin`, únicamente se elimina la ACL que especifica `/bin/login` como sujeto y `/bin` como el objeto.

**ADVERTENCIA: Especificando la opción `-Z` o `-D` sin argumento, elimina todas las reglas ACL.**

## **Un buen esquema de protección de archivos y bases de datos**

En esta sección le mostraré un buen esquema de protección que puede utilizar con LIDS. Este esquema le permite convertir al directorio `/boot` (o partición) en un directorio de sólo lectura, lo que significa que los intrusos no pueden modificar el kernel; también convierte los directorios `/lib`, `/root`, `/etc`, `/sbin` y `/usr/sbin`, `/usr/bin` y `/bin` en directorios de sólo lectura. Además permite operaciones del modo `append` para los archivos del directorio `/var/log`, lo que garantiza que los archivos de registro no son destruidos por intrusos. Esta configuración se muestra a continuación:

```
# Convertir el directorio /boot o la partición de solo lectura  
/sbin/lidsadm -A -o /boot -j READ  
  
# Convertir el directorio library del sistema de solo lectura  
# Esto protege también a lib/modules  
/sbin/lidsadm -A -o /lib -j READ  
  
# Convertir el directorio local del usuario root en solo  
lectura  
/sbin/lidsadm -A -o /root -j READ  
  
# Convertir el directorio de configuración en solo lectura  
/sbin/lidsadm -A -o /etc -j READ  
  
# Convertir el directorio binario del demonio en solo lectura  
/sbin/lidsadm -A -o /sbin -j READ  
  
# Convertir el otro directorio binario del demonio en solo  
lectura  
/sbin/lidsadm -A -o /usr/sbin -j READ  
  
# Convertir el directorio general de binarios en solo lectura  
/sbin/lidsadm -A -o /bin -j READ
```

```

# Convertir el otro directorio general de binarios en sólo lectura
/sbin/lidsadm -A -o /usr/bin -j READ

# Convertir el directorio general library en sólo lectura
/sbin/lidsadm -A -o /usr/lib -j READ

# Convertir el directorio de registros del sistema en el modo
append
/sbin/lidsadm -A -o /var/log -j APPEND

# Convertir el directorio de binarios de X Windows en sólo lectura
/sbin/lidsadm -A -o /usr/X11R6/bin -j READ

```

Además de proteger sus archivos y directorios, utilizando la técnica anterior, LIDS puede utilizar las capacidades Linux para limitar las capacidades de un programa en ejecución (es decir, en proceso). En un sistema Linux tradicional, el usuario root (es decir, el usuario con UID y GID con el valor 0) tiene todas las "capacidades" o la habilidad de realizar cualquier tarea ejecutando cualquier proceso. LIDS utiliza las capacidades Linux para romper el poder del root (o de los procesos que ejecuta el usuario root) en partes en las que pueda especificar adecuadamente procesos determinados que pueden o no tener lugar. Para encontrar más información sobre las capacidades Linux disponibles, observe el archivo de cabecera `/usr/include/linux/capability.h`. La tabla 23.5 contiene una lista de todas las capacidades Linux y de su estado (activado o desactivado) en el archivo de configuración por defecto de las capacidades LIDS, `/etc/lids/lids.cap`.

**Tabla 23.5.** Lista de las capacidades de Linux

ID capacidad	Nombre de la capacidad	Estado en /etc/lids/lids.cap	Significado
0	CAP_CHOWN	Activado	Activa/desactiva el cambio de dueño del archivo.
1	CAP_DAC_OVERRIDE	Activado	Activa/desactiva la invalidación de todas las restricciones de acceso DAC.
2	CAP_DAC_READ_SEARCH	Activado	Activa/desactiva la invalidación de todas las restricciones de lectura y búsqueda DAC.
3	CAP_FOWNER	Activado	Activa/desactiva las restricciones siguientes: el ID del usuario

ID capacidad	Nombre de la capacidad	Estado en /etc/ lids/lids.cap	Significado
			efectivo debería coincidir con el ID del dueño cuando se asignan los bits S_ISUID y S_ISGID en un archivo; el ID del grupo efectivo debe coincidir con el ID del dueño cuando se asigna ese bit en un archivo.
4	CAP_FSETID	Activado	Activa/desactiva el acceso cuando el ID del usuario efectivo no es igual al ID del dueño.
5	CAP_KILL	Activado	Activa/desactiva el envío de señales a procesos que pertenecen a otros.
6	CAP_SETGID	Activado	Activa/desactiva el cambio de GID.
7	CAP_SETUID	Activado	Activa/desactiva el cambio de UID.
8	CAP_SETPCAP	Activado	Activa/desactiva la transferencia y la eliminación del conjunto actual a cualquier PID.
9	CAP_LINUX_IMMUTABLE	Desactivado	Activa/desactiva la modificación de archivos inmutables de modo append.
10	CAP_NET_BIND_SERVICE	Desactivado	Activa/desactiva la conexión a puertos pertenecientes a 1024.
11	CAP_NET_BROADCAST	Activado	Activa/desactiva la transmisión / escucha de una multidifusión.
12	CAP_NET_ADMIN	Desactivado	Activa/desactiva la posibilidad de que la administración de

ID capacidad	Nombre de la capacidad	Estado en /etc/ lids/lids.cap	Significado
			red realice configuraciones de interfaces, administre firewalls IP, establezca enmascaramiento, establezca cuentas IP, establezca opciones de depuración en los sockets, modifique tablas de enrutamiento, establezca procesos arbitrarios de dueños de grupos en sockets, conecte cualquier dirección a un proxy transparente, establezca servicio Type Of Service (TOS), establezca modo múltiple, etc.
13	CAP_NET_RAW	Desactivado	Activa/desactiva la utilización de sockets.
14	CAP_IPC_LOCK	Activado	Activa/desactiva el bloqueo de segmentos de memoria compartida.
15	CAP_IPC_OWNER	Activado	Activa/desactiva comprobación de dueños de IPC.
16	CAP_SYS_MODULE	Desactivado	Activa/desactiva inserción y eliminación de módulos del kernel.
17	CAP_SYS_RAWIO	Desactivado	Permite que ioperm (2)/iopl(2) accedan a CAP_SYS_CHROOT chroot (2).
18	CAP_SYS_CHROOT	Desactivado	Activa/desactiva llamadas al sistema chroot.
19	CAP_SYS_PTRACE	Activado	Activa/desactiva ptrace.
20	CAP_SYS_PACCT	Activado	Activa/desactiva la configuración de contabilidad de procesos.

ID capacidad	Nombre de la capacidad	Estado en /etc/lids/lids.cap	Significado
21	CAP_SYS_ADMIN	Activado	Activa/desactiva distintas tareas de administración del sistema.
22	CAP_SYS_BOOT	Activado	Activa/desactiva reboot.
23	CAP_SYS_NICE	Activado	Activa/desactiva el cambio de prioridad de procesos utilizando el comando nice.
24	CAP_SYS_RESOURCE	Activado	Activa/desactiva la asignación del límite de recursos del sistema.
25	CAP_SYS_TIME	Activado	Activa/desactiva la asignación de la hora del sistema.
26	CAP_SYS_TTY_CONFIG	Activado	Activa/desactiva la configuración seudoterminal (TTY).
27	CAP_MKNOD	Activado	Activa/desactiva los aspectos privilegiados de la llamada al sistema mknod().
28	CAPLEASE	Activado	Activa/desactiva la renovación de archivos.
29	CAPHIDDEN	Activado	Activa/desactiva la ocultación de un proceso al resto del sistema.
30	CAP_INIT_KILL	Activado	Activa/desactiva la capacidad de los programas para matar hijos del proceso init (PID = 1).

Las asignaciones por defecto para las capacidades de Linux están almacenadas en el archivo /etc/lids/lids.cap, tal y como se muestra en el listado 23.8. El signo + activa la capacidad y el signo – la desactiva. Por ejemplo, en la lista anterior, la última capacidad de Linux llamada CAP\_INIT\_KILL está activada, lo que significa que un proceso propio del root podría matar cualquier

proceso hijo (normalmente demonios) creado por el proceso `init`. Puede utilizar un editor de texto para activar o desactivar las capacidades Linux.

**Listado 23.8. /etc/lids/lids.cap**

```
+0:CAP_CHOWN  
+1:CAP_DAC_OVERRIDE  
+2:CAP_DAC_READ_SEARCH  
+3:CAP_FOWNER  
+4:CAP_FSETID  
+5:CAP_KILL  
+6:CAP_SETGID  
+7:CAP_SETUID  
+8:CAP_SETPCAP  
-9:CAP_LINUX_IMMUTABLE  
-10:CAP_NET_BIND_SERVICE  
+11:CAP_NET_BROADCAST  
-12:CAP_NET_ADMIN  
-13:CAP_NET_RAW  
+14:CAP_IPC_LOCK  
+15:CAP_IPC_OWNER  
-16:CAP_SYS_MODULE  
-17:CAP_SYS_RAWIO  
-18:CAP_SYS_CHROOT  
+19:CAP_SYS_PTRACE  
+20:CAP_SYS_PACCT  
-21:CAP_SYS_ADMIN  
+22:CAP_SYS_BOOT  
+23:CAP_SYS_NICE  
+24:CAP_SYS_RESOURCE  
+25:CAP_SYS_TIME  
+26:CAP_SYS_TTY_CONFIG  
+27:CAP_MKNOD  
+28:CAPLEASE  
+29:CAP_HIDDEN  
+30:CAP_INIT_KILL
```

## **Proteger su sistema utilizando las capacidades de Linux gestionadas por LIDS**

Puede utilizar las capacidades proporcionadas por LIDS para proteger su sistema. En esta sección, le mostraré cómo sacar partido de las capacidades Linux manejadas por LIDS. Verá cómo puede proteger demonios (como el servidor Web de Apache) para que no sean asesinados por el usuario root, cómo ocultar procesos a los programas como `ps`, cómo desactivar acceso a dispositivos y cómo proteger el indicador inmutable `ext2`.

### **Proteger demonios para que el root no los asesine**

Normalmente, los procesos demonio, como el agente de transporte de correo Sendmail y el servidor Web Apache, son iniciados por el proceso `init`. Si quiere

protegerlos para que el usuario root no pueda asesinarlos, modifique las opciones CAP\_INIT\_KILL en /etc/lids/lids.cap:

```
-30:CAP_INIT_KILL
```

Cuando haya vuelto a cargar la configuración LIDS (utilizando el comando /sbin/lidsadm -S -- + RELOAD\_CONF) o reiniciando el sistema y cerrado el kernel (utilizando el comando /sbin/lidsadm -I en el script /etc/rc.d/rc.local), usted (como root) no será capaz de asesinar a los hijos init. Esto garantiza que aunque su sistema esté comprometido y un intruso obtenga privilegios de root, el intruso no puede asesinar los demonios y reemplazarlos con versiones Trojanas.

## Ocultar procesos

Por defecto, la capacidad CAP\_HIDDEN está activada en el archivo de configuración /etc/lids/lids.cap. Puede ocultar un proceso a todo el mundo utilizando el comando siguiente:

```
lidsadm -A -s /path/to/binary -t -o CAP_HIDDEN -j INHERIT
```

donde /path/to/binary es la ruta completa del ejecutable que quiere ocultar durante la ejecución. Por ejemplo, para ocultar el proceso del servidor Apache /usr/local/apache/bin/httpd durante su ejecución, simplemente ejecute el comando siguiente:

```
lidsadm -A -s /usr/local/apache/bin/httpd -t -o CAP_HIDDEN -j  
INHERIT
```

Esto etiqueta al proceso como proceso oculto en el kernel y no se puede encontrar utilizando ninguna herramienta descargada por el usuario como ps o top, o incluso explorar archivos en el sistema de archivos /proc.

## Desactivar el acceso de los procesos a dispositivos

Normalmente, sólo los procesos especiales necesitan acceder a los dispositivos. Por lo que es una buena idea desactivar el acceso a dispositivos y activar el acceso a ellos sólo si es necesario, lo que está de acuerdo con el concepto general de seguridad de "cierre todo, abra sólo lo que necesita."

El acceso a dispositivos se controla con la capacidad CAP\_SYS\_RAWIO, que está desactivada por defecto en el archivo de configuración /etc/lids/lids.cap. Si estuviese activada, los procesos podrían acceder a ioperm/iopi, /dev/port, /dev/mem, /dev/kmem y a otros dispositivos de bloque. Cuando esta capacidad está desactivada (como está por defecto) el programa /sbin/lilo no puede funcionar de forma adecuada porque necesita acceso al nivel de los dispositivos en el disco duro.

Pero algunos programas especiales, como XF86\_SVGA, podrían necesitar que esta capacidad se ejecutase adecuadamente. En ese caso, puede añadir el programa a la lista de excepciones, del siguiente modo:

```
lidsadm -A -s /usr/X11R6/bin/XF86_SVGA -t -o CAP_SYS_RAWIO -j  
INHERIT
```

Esto le da a XF86\_SVGA la capacidad de CAP\_SYS\_RAWIO, mientras que otros programas son incapaces de obtener capacidad CAP\_SYS\_RAWIO.

## Desactivar tareas de administración de red

Por defecto, la capacidad CAP\_NET\_ADMIN está desactivada, por lo que un administrador de red (normalmente el usuario root) no puede seguir realizando las siguientes tareas de administración del sistema:

- Configurar la interfaz Ethernet
- Administrar el firewall IP, enmascaramiento y contabilidad
- Determinar la opción de depuración en los sockets
- Modificar tablas de enrutamiento
- Asignar dueños arbitrarios de procesos en los sockets
- Conectarse a cualquier dirección para proxy transparente
- Asignar Type of Service (TOS)
- Asignar modo múltiple
- Despejar estadísticas sobre las unidades de disco
- Multidifusión
- Asignar la lectura / escritura de registros específicos de dispositivos

Se recomienda la asignación por defecto. Si tiene que realizar alguna de las tareas anteriores, simplemente desactive LIDS temporalmente, con el comando /sbin/lidsadm -S -- -LIDS.

## Proteger archivos inmutables

El sistema de archivos ext2 tiene una característica ampliable que le permite indicar un archivo como inmutable. Esto se realiza con el comando chattr. Por ejemplo, chattr +i /path/to/myfile convierte a /path/to/myfile en un archivo inmutable.

Un archivo que tenga el atributo inmutable no se puede modificar, ni borrar, ni se le puede cambiar el nombre, ni puede enlazarse simbólicamente. Sin embargo, el usuario root es capaz de cambiar el indicador utilizando el comando chattr -i /path/to/myfile. Puede proteger archivos inmutables incluso del usuario root, desactivando la capacidad CAP\_LINUX\_IMMUTABLE. Observe que CAP\_LINUX\_IMMUTABLE está desactivada por defecto en /etc/lids/lids.cap.

## Detectar un escáner

Si tiene activado el escáner de puertos integrado durante la compilación del kernel, tal y como se recomienda en la sección de instalación de LIDS, puede detectar el escaneamiento de puertos. Este escáner puede detectar un escáner medio abierto, un escáner de puertos stealth SYN, Stealth FIN, Xmas, o un escáner Null, entre otros. Las herramientas como nmap y Satan o Saint, se pueden detectar con un detector. Esto resulta de utilidad cuando está desactivado el socket (CAP\_NET\_RAW).

**NOTA:** Cuando se desactiva CAP\_NET\_RAW algunos escáneres cargados por el usuario, basados en redirección, no van a funcionar adecuadamente. Pero el escáner suministrado en el kernel proporcionado por LIDS, no utiliza ningún socket, que lo hace más seguro. Podría utilizar el escáner suministrado por LIDS.

## Responder a un intruso

Cuando LIDS detecta una violación de cualquiera de las reglas ACL, puede responder utilizando el método siguiente.

- **Registrar el mensaje.** Cuando viola una regla ACL, LIDS registrará un mensaje utilizando el demonio kernel de registro (klogd).
- **Enviar un correo electrónico a la autoridad adecuada.** LIDS puede enviar un correo electrónico cuando tiene lugar una violación de alguna de las reglas. Esta característica está controlada mediante el archivo /etc/lids/lids.net, que se discute en el paso 5 de la sección "Compilar, instalar y configurar LIDS" de este capítulo.
- **Ejecutar la consola.** Si activa esta opción durante el parcheo del kernel de LIDS, tal y como se vio en el paso 9 de la sección de instalación de LIDS, la consola se lanzará cuando un usuario viole una regla ACL.

Un sistema parecido al LIDS es el proyecto OpenWall ([www.openwall.com/linux](http://www.openwall.com/linux)). El proyecto OpenWall tiene algunas características distintas de las que encontramos en LIDS, y un parche concreto de OpenWall fabrica el área de amontonamiento de un proceso no ejecutable. Debería remitirse al proyecto de este trabajo en proceso.

# **Parte VII**

# **Apéndices**



# A Códigos de estado HTTP 1.1

---

Para cada solicitud de un cliente Web, el servidor Web debe devolver un código de estado HTTP al cliente, que consiste en un número de tres dígitos. Un cliente Web puede intentar entender la respuesta del servidor observando estos códigos de estado, enviados en la cabecera Status-Line HTTP. El código llega acompañado de una frase corta, llamada frase motivo, que intenta proporcionar una breve explicación al usuario. Por ejemplo, una cabecera Status-Line HTTP podría ser como esta:

```
HTTP/1.1 404 Not Found
```

Aquí, 404 es el código de estado, y Not Found es la frase motivo. Un típico cliente Web, como un navegador Web, mostrará la frase Not Found en la ventana del navegador. Hay cinco tipos distintos de códigos de estado desde las últimas especificaciones de HTTP 1.1; estos tipos de códigos se discuten en las secciones siguientes.

## Códigos de estado de información (100–199)

El propósito de este tipo de códigos de estado es que el cliente sepa que el servidor está procesando una solicitud. Estos códigos de estado son sólo de tipo

informativo; el cliente no tiene que actuar sobre ellos. Tenga en cuenta que HTTP 1.0 no define códigos de estado 1xx, por lo que los códigos de estado 1xx no se deben enviar a los clientes HTTP 1.0. Los códigos de estado 1xx definidos actualmente son:

100 Continue: el servidor envía este código para que el cliente sepa que está preparado para recibir el resto de la solicitud.

101 Switching Protocols: el servidor envía este código cuando está dispuesto a cambiar el protocolo de aplicación a uno especificado en una cabecera de una solicitud Upgrade, proporcionada por el cliente. El cambio sólo tendrá lugar si el nuevo protocolo proporciona alguna ventaja sobre el protocolo existente. Por ejemplo, el cliente podría solicitar que el servidor utilice un protocolo HTTP más moderno que el que se está utilizando. En tal caso, el servidor debería cambiarlo si es posible.

## Éxito en la solicitud del cliente (200–299)

Un código devuelto en el rango de 200–299, indica que la solicitud del cliente se ha recibido y se ha aceptado. Los códigos de estado 2xx definidos actualmente son:

200 OK: el servidor tiene éxito en el procesamiento de la solicitud, y se adjunta el documento solicitado.

201 Created: el servidor crea una URI nueva, especificada en una cabecera Location.

202 Accepted: se aceptó la solicitud para ser procesada, pero el servidor no ha completado aún el proceso.

203 Non-Authoritative Information: la meta información en la cabecera de respuesta no se origina en el servidor sino que se copia de otro servidor.

204 No Content: la solicitud está completa, pero no se necesita nueva información para devolverla. El cliente debería continuar mostrando el documento actual.

205 Reset Content: el cliente debería reajustar el documento actual. Esto es de utilidad cuando hay que reajustar un formulario HTML para borrar los valores existentes en los campos de entrada.

206 Partial Content: el servidor tiene cumplimentado la solicitud GET parcial para el recurso. Este código se utiliza para responder solicitudes Range. El servidor envía una cabecera Content-Range para indicar qué segmentos de datos están adjuntos.

# Redirección de solicitudes (300–399)

Los códigos de estado en el rango 300–399 se envían al cliente para que conozca la necesidad de realizar una serie de acciones para completar la solicitud. Los códigos de estado 3xx definidos actualmente son:

**300 Multiple Choices:** los recursos solicitados corresponden a un conjunto de documentos. El servidor puede enviar información sobre cada documento con su propia información sobre las localizaciones específicas y sobre la negociación del contenido, de modo que el cliente pueda elegir uno.

**301 Moved Permanently:** el recurso solicitado no existe en el servidor. Se envía una cabecera Location para redirigir al cliente a la nueva URL. El cliente dirige todas sus solicitudes futuras a la nueva URI.

**302 Moved Temporarily:** los recursos solicitados se han movido temporalmente. Se envía una cabecera Location para redirigir al cliente a la nueva URL. El cliente sigue utilizando la URI antigua en sus futuras solicitudes.

**303 See Other:** el recurso solicitado se encuentra en una localización distinta a la indicada por la cabecera Location, y el cliente debería utilizar el método GET para recuperarlo.

**304 Not Modified:** el servidor utiliza este código para responder a la cabecera de la solicitud If-Modified-Since. Eso indica que el documento solicitado no ha sido modificado desde la fecha especificada, y que el cliente debería utilizar la copia del caché.

**305 Use Proxy:** el cliente debería utilizar un proxy, especificado por la cabecera Location, para recuperar el recurso solicitado.

**307 Temporary Redirect:** el recurso solicitado está, temporalmente, redirigido a una localización diferente. Se envía una cabecera Location para redirigir al cliente a la nueva URL. El cliente sigue utilizando la URI antigua en sus futuras solicitudes.

# Solicitud del cliente incompleta (400–499)

Los códigos de estado en el rango 400–499 se envían para indicar que la solicitud del cliente está incompleta y que se necesita más información para completar la solicitud del recurso. Los códigos de estado 4xx definidos actualmente son:

**400 Bad Request:** el servidor detecta un error de sintaxis en la solicitud del cliente.

401 Unauthorized: la solicitud necesita autenticación del usuario. El servidor envía una cabecera WWW-Authenticate para indicar el tipo de autenticación y el campo del recurso solicitado.

402 Payment Required: este código está reservado para futuros usos.

403 Forbidden: el acceso al recurso solicitado está prohibido. El cliente no debería repetir la solicitud.

404 Not Found: el documento solicitado no existe en el servidor.

405 Method Not Allowed: el método de la solicitud utilizado por el cliente. El servidor envía la cabecera Allow indicando qué métodos son aceptables para acceder al recurso solicitado.

406 Not Acceptable: el recurso solicitado no está disponible en un formato que pueda aceptar el cliente, basado en las cabeceras Accept recibidas en el servidor. Si la solicitud no era una solicitud HEAD, el servidor puede enviar cabeceras Content-Language, Content-Encoding y Content-Type para indicar qué formatos están disponibles.

407 Proxy Authentication Required: acceso no autorizado al servidor proxy. El cliente debe autenticarse primero con el proxy. El servidor envía la cabecera Proxy-Authenticate indicando el esquema de autenticación y el campo del recurso solicitado.

408 Request Time-Out: el cliente tiene que fallar para completar su solicitud dentro del período timeout utilizado por el servidor. Sin embargo, el cliente puede repetir la solicitud.

409 Conflict: la solicitud del cliente está en conflicto con otra solicitud. El servidor puede añadir información sobre el tipo de conflicto junto con el código de estado.

410 Gone: el recurso solicitado está permanente fuera del servidor.

411 Length Required: el cliente debe suministrar una cabecera Content-Length en su solicitud.

412 Precondition Failed: cuando un cliente envía una solicitud con una o más cabeceras If..., el servidor utiliza este código para indicar que una o más de las condiciones especificadas en esta cabecera son falsas.

413 Request Entity Too Large: el servidor rechaza procesar la solicitud porque el cuerpo del mensaje es demasiado largo. El servidor puede cerrar la conexión para que el cliente no continúe con la solicitud.

414 Request-URI Too Long: el servidor rechaza procesar la solicitud porque la URI especificada es demasiado larga.

415 Unsupported Media Type: el servidor rechaza procesar la solicitud porque no soporta el formato del cuerpo del documento.

417 Expectation Failed: el servidor falla en los requisitos de la cabecera Expect Request.

## Errores del servidor (500-599)

Los códigos de estado en el rango 500–599 se devuelven cuando el servidor encuentra un error y no puede cumplir con la solicitud. Los códigos de estado 5xx definidos actualmente son los siguientes:

500 Internal Server Error: una configuración del servidor o un programa externo ha causado un error.

501 Not Implemented: el servidor no soporta la funcionalidad necesaria para cumplir con la solicitud.

502 Bad Gateway: el servidor encuentra una respuesta inválida desde un servidor superior o un proxy.

503 Service Unavailable: el servicio no está disponible temporalmente. El servidor puede enviar una cabecera Retry-After para indicar cuándo podría estar de nuevo disponible el servicio.

504 Gateway Time-Out: el gateway o el proxy tienen problemas de conexión.

505 HTTP Version Not Supported: no soporta la versión de HTTP que utiliza el cliente.



# B Entender las expresiones regulares

---

Una expresión regular está compuesta normalmente por caracteres normales y especiales para crear un patrón. El patrón se utiliza para que corresponda a una o más subcadenas o para que coincida con una cadena completa o string. Por ejemplo:

([a-z]+) \. ([a-z]) \. ([a-z]+)

es una expresión regular que corresponde a [www.idgbooks.com](http://www.idgbooks.com), [www.apache.org](http://www.apache.org), y similares. Los caracteres especiales utilizados en una expresión regular se suelen llamar caracteres meta. La tabla B.1 muestra los caracteres meta utilizados habitualmente.

**Tabla B.1.** Caracteres meta utilizados habitualmente

Carácter meta	Función
.	Corresponde a cualquier carácter (excepto al carácter de línea nueva).
^	Corresponde al inicio del string.
\$	Corresponde al final del string.

Carácter meta	Función
\b	Corresponde al límite de la palabra.
x?	Corresponde a 0 o a 1 veces x, donde x es cualquier expresión regular.
x*	Corresponde a 0 o más veces x.
x+	Corresponde a 1 o más veces x.
foo bar	Corresponde a foo o a bar.
[abc]	Corresponde a cualquier carácter en el conjunto abc.
[A-Z]	Corresponde a cualquier carácter en el rango de la A a la Z.
[^xyz]	Corresponde a cualquier carácter que no esté en el conjunto xyz.
\w	Corresponde a un carácter alfanumérico (por ejemplo, [a-zA-Z0-9_]).
\s	Corresponde a un carácter de espacio en blanco.
\t	Carácter de tabulación.
\n	Carácter de nueva línea.
\r	Carácter retorno de carro.
\f	Carácter salto de página.
\v	Tabulación vertical.
\a	Carácter Bell.
\e	Carácter de escape.
\077	Char Octal.
\x9f	Char Hex.
\c[	Char de control.
\l	El siguiente char en minúsculas.
\L	Minúsculas till \E.
\U	Mayúsculas till \E.
\E	Modificación letra final.
\Q	El siguiente char en mayúsculas.
\u	Carácter meta para comillas till \E.

Si tiene que utilizar un carácter meta como si fuera carácter normal en una expresión regular, puede utilizar el formato \metachar para salir del significa-

do especial. Un ejemplo de esto es \\$, que es el carácter regular del signo del dólar. La tabla B.2 muestra los multiplicadores estándar utilizados en las expresiones regulares.

**Tabla B.2.** Multiplicadores estándar

Multiplicador	Significado
.	Corresponde a 0 o más veces.
+	Corresponde a 1 o más veces.
?	Corresponde a 1 o 0 veces.
{n}	Corresponde exactamente a n veces.
{n,}	Corresponde al menos a n veces.
{n, m}	Corresponde al menos a n veces pero a no menos de m veces.

El carácter A | se trata como un operador OR. Un par de paréntesis () le permiten definir el área utilizada para la búsqueda de coincidencias en una expresión regular. Un par de corchetes [] crean una clase o un rango.

Vamos a volver al primer ejemplo:

([a-z]+)\.([a-z])\.( [a-z]+)

Como se mencionó antes, esta expresión se puede utilizar para hacer corresponder strings como www.hungryminds.com. El primer [a-z]+ especifica que son necesarios uno o más caracteres en el rango de la a a la z para que coincida el grupo especificado por el primer par de paréntesis. Si se encuentra correspondencia, se podrá acceder a ella utilizando \$1. Hay tres pares de paréntesis en esta expresión. El primer par (empezando por la izquierda) es \$1, el segundo es \$2, y el tercero es \$3. Observe que \ se utiliza para escapar del carácter meta del punto(.) entre los grupos.

A continuación, tiene dos ejemplos:

- ^foo\.htm\$: corresponde al string foo.htm. No corresponde a afoo.htm porque el carácter meta ^ se utiliza para especificar que el string debe comenzar con el carácter f. Tampoco se corresponde con foo.html porque el carácter meta \$ se utiliza para especificar que el string debe terminar con el carácter m.
- ^www\.( [^.]+)\.host\..com(.\*): corresponderá a un string del tipo www.username.host.com STATUS=java y \$1 se asignará al host y \$2 sostendrá todo lo que va a continuación de www.username.host.com del string. El \$2 contiene STATUS=java.



# C Recursos Apache online

---

Este apéndice contiene una lista de los sitios Web, grupos Usenet y listas de correo relacionados con Apache, que pueden resultarle de utilidad.

## Recursos gratuitos

Al igual que Apache, muchos de los mejores recursos para Apache son gratuitos en Internet. Las siguientes secciones describen algunos de estos recursos gratuitos de Internet para Apache.

### Sitios Web

A continuación tiene algunos de los mejores sitios Web Apache:

Official Apache Web site: [www.apache.org](http://www.apache.org)

Apache Module Registry: <http://modules.apache.org>

Apache/Perl Integration Project: <http://perl.apache.org>

Apache-SSL: [www.apache-ssl.org](http://www.apache-ssl.org)

Jakarta Project: <http://jakarta.apache.org>

Apache GUI Project: <http://gui.apache.org>

Apache Today: [www.apachetoday.com](http://www.apachetoday.com)

Apache Week: [www.apacheweek.com](http://www.apacheweek.com)

## Grupos de noticias Usenet

Los grupos de noticias Usenet son un excelente recurso para cualquiera que esté interesado en aprender de experiencias personales y de expertos de todo el mundo. Cuando tiene un problema con Apache, un script CGI, o un servlet de Java, hay posibilidades de que alguien se haya encontrado ya con ese problema. Buscar respuestas en Usenet es un procedimiento habitual para la mayoría de los administradores de sistemas. Si tiene acceso a un servicio Usenet mediante su ISP, entonces puede acceder a cualquier grupo de noticias público. Si no tiene acceso a Usenet mediante su ISP, siempre puede utilizar Google Groups en <http://groups.google.com>. Google Groups le permite navegar por todos los grupos de noticias Usenet y participar en ellos. Si quiere realizar búsquedas complejas, puede utilizar [http://groups.google.com/advanced\\_group\\_search](http://groups.google.com/advanced_group_search). Evite crear discusiones innecesarias, proponiendo preguntas del tipo "¿es mejor IE que Netscape Navigator?", ya que cada persona va a tener sus preferencias, y normalmente se adentran en el análisis objetivo de este tipo de asuntos. Realice preguntas constructivas y no olvide contestar si puede hacerlo.

## Grupos de noticias relacionados con servidores Web

Estos grupos de noticias son un recurso muy bueno para obtener información sobre distintos servidores Web. Puede realizar preguntas o seguir discusiones sobre Apache en un grupo específico de plataforma. Por ejemplo, si está ejecutando Apache en Windows, puede realizar preguntas en el grupo: `comp.infosystems.www.servers.ms-windows`.

### **comp.infosystems.www.servers.unix**

Este grupo de noticias discute sobre servidores Web para plataformas UNIX. Incluye asuntos como preguntas y soluciones sobre configuración, aspectos de seguridad, estructura de directorios e informes de depuración.

### **comp.infosystems.www.servers.ms-windows**

Este grupo de noticias cubre servidores Web para las plataformas MS Windows y NT. Incluye asuntos como preguntas y soluciones sobre configuración, aspectos de seguridad, estructura de directorios e informes de depuración.

### **comp.infosystems.www.servers.mac**

Este grupo de noticias sostiene discusiones sobre servidores Web para la plataforma Macintosh (Mac OS). Incluye asuntos como preguntas y soluciones so-

bre configuración, aspectos de seguridad, estructura de directorios e informes de depuración.

### **comp.infosystems.www.servers.misc**

Este grupo de noticias discute servidores Web para otras plataformas como Amiga y VMS. Incluye asuntos como preguntas y soluciones sobre configuración, aspectos de seguridad, estructura de directorios e informes de depuración.

## **Grupos de noticias relacionados con lenguajes de autor**

Los grupos de noticias de esta categoría tratan con herramientas y técnicas de autor de contenido Web. Puede realizar preguntas o seguir discusiones sobre programación CGI, etiquetas HTML, imágenes, y cosas de este tipo.

### **comp.infosystems.www.authoring.cgi**

Este grupo de noticias discute el desarrollo de scripts de Common Gateway Interface (CGI) en cuanto a su relación con páginas Web de autor. Se incluyen asuntos como el manejo de los resultados de un formulario, cómo crear imágenes al vuelo y cómo fusionar otras ofertas Web interactivas.

### **comp.infosystems.www.authoring.html**

Este grupo de noticias discute el lenguaje HyperText Markup Language (HTML) en cuanto a su relación con páginas Web de autor. Se incluyen aspectos del tipo editores HTML, trucos para formatear y estándares HTML actuales y propuestos.

### **comp.infosystems.www.authoring.images**

Este grupo de noticias discute la creación y edición de imágenes en cuanto a su relación con páginas Web de autor. Los asuntos posibles incluyen cómo influir en las capacidades de despliegue de imágenes en la Web y preguntas y respuesta habituales sobre el soporte de mapas de imágenes.

### **comp.infosystems.www.authoring.misc**

Este grupo de noticias cubre una mezcla de asuntos sobre páginas Web de autor que no cubre el resto de los grupos `comp.infosystems.www.authoring.*`. Se incluyen aspectos de audio y vídeo.

### **comp.infosystems.www.authoring.site-design**

Estos grupos de noticias cubren aspectos sobre el diseño de sitios. Puede aprender sobre diseños buenos y malos que han utilizado otras personas.

### **comp.infosystems.www.authoring.stylesheets**

Este grupo de noticias cubre hojas de estilo que se utilizan en el desarrollo de páginas Web.

## **comp.infosystems.www.authoring.tools**

Este grupo de noticias discute sobre herramientas Web, pasando por herramientas de autor que le permiten crear contenidos Web para ingenierías de publicación Web a gran escala.

## **Grupos de noticias relacionados con navegadores Web**

Si está interesado en los navegadores Web en las distintas plataformas, puede navegar o participar en los grupos de noticias de esta categoría.

### **comp.infosystems.www.browsers.ms-windows**

Este grupo de noticias discute sobre navegadores Web para las plataformas Windows y NT. Se incluyen aspectos como preguntas y soluciones sobre configuración, visores externos (aplicaciones de ayuda) e informes de depuración.

### **comp.infosystems.www.browsers.mac**

Estos grupos de noticias discuten sobre navegadores Web para la plataforma Macintosh. Se incluyen aspectos como preguntas y soluciones sobre configuración, visores externos (aplicaciones de ayuda) e informes de depuración.

### **comp.infosystems.www.browsers.x**

Estos grupos de noticias discuten sobre navegadores Web para el sistema Windows X. Se incluyen aspectos como preguntas y soluciones sobre configuración, visores externos (aplicaciones de ayuda) e informes de depuración.

### **comp.infosystems.www.browsers.misc**

Estos grupos de noticias discuten sobre navegadores Web para el resto de plataformas. Se incluyen aspectos como preguntas y soluciones sobre configuración, visores externos (aplicaciones de ayuda) e informes de depuración. Incluye plataformas como Amiga, DOS, VMS y las de modo de texto de Unix.

## **Grupos de noticias de anuncios**

Hay varios grupos que utilizan esta categoría de grupos de noticias para anunciar sus lanzamientos de software, alertas de seguridad en cuanto a software Web y otros asuntos de importancia. No debe participar en ninguno de estos grupos a no ser que tenga información relacionada con el grupo.

### **comp.infosystems.www.announce**

Este es un grupo de noticias en el que se comunican nuevos recursos relacionados con la Web.

## **Otros grupos de noticias WWW**

Los grupos de noticias de esta categoría son grupos que tratan asuntos relacionados con la Web que no caben en otra categoría.

## **comp.infosystems.www.advocacy**

Este grupo de noticias se ha creado para comentarios, argumentos y debates sobre navegadores, servidores, visores externos, y otro tipo de software.

## **comp.infosystems.www.misc**

Proporciona un foro para discusiones generales sobre asuntos relacionados con WWW que no se cubren en otros grupos de noticias. Este grupo va a incluir, probablemente, discusiones sobre cambios futuros en la estructura Web y sobre protocolos de la Web que afectan tanto a clientes como a servidores.

## **Grupos de noticias Perl**

Como Perl se utiliza en la programación CGI, mod\_perl y FastCGI, los siguientes grupos de noticias le pueden resultar de gran utilidad.

### **comp.lang.perl.misc**

Este grupo de noticias discute sobre Perl; incluye todo lo relacionado con Perl, desde informes de depuración sobre nuevas características hasta historias y anécdotas. Es la mejor fuente para informarse sobre las novedades en Perl.

### **comp.lang.perl.announce**

Se anuncian nuevos lanzamientos, FAQ y nuevos módulos.

## **Listas de correo**

Si quiere recibir noticias Apache sobre la fundación Apache y sus actividades, suscríbase a announce-subscribe@apache.org. Recuerde que no es una lista de correo de discusión. Simplemente es utilizada por la fundación Apache para anunciar asuntos de interés periodístico. Por favor no realice preguntas ni haga comentarios en este sitio. Si está interesado en conferencias sobre Apache, las cuales constituyen un excelente recurso para los administradores Apache serios, puede suscribirse a la lista announce-subscribe@apachecon.com list. También es una lista de anuncios y no es adecuada para participar con preguntas o comentarios. Si está interesado en asuntos relacionados con el nuevo desarrollo Apache, entonces puede suscribirse a new-httdp-subscribe@apache.org. También puede suscribirse a http://groups.yahoo.com/group/new-httdp, que es un grupo Yahoo! dedicado al servidor Apache.

## **Recursos comerciales**

Un número creciente de usuarios Apache (principalmente usuarios corporativos) está continuamente buscando recursos comerciales que ofrezcan software o servicios Apache. Algunos de estos recursos comerciales para Apache son:

**Stronghold:** [www.c2.net](http://www.c2.net)

**Covalent Raven:** <http://raven.covalent.net>

**Rovis:** [www.rovis.com/warpaint/](http://www.rovis.com/warpaint/)

## Otros recursos relacionados

Hay muchos sitios Web que pueden ser de utilidad para un desarrollador Web o para un administrador. A continuación tiene una lista de recursos.

**WWW Consortium:** [www.w3.org/](http://www.w3.org/)

**Netcraft Survey Report Web site:** [www.netcraft.co.uk/Survey/](http://www.netcraft.co.uk/Survey/)

**Server Watch:** [www.serverwatch.com](http://www.serverwatch.com)

**Search Engine Watch:** [www.searchenginewatch.com](http://www.searchenginewatch.com)

**Browser Watch:** [www.browserwatch.com](http://www.browserwatch.com)

**Web Compare:** [www.webcompare.com](http://www.webcompare.com)

**Web Developer:** [www.webdeveloper.com](http://www.webdeveloper.com)

**Web Reference:** [www.webreference.com](http://www.webreference.com)

**Electronic Commerce on Internet:** <http://e-comm.internet.com>

**ISP Buyer's Guide:** [www.TheList.com](http://www.TheList.com)

**Internet News:** [www.InternetNews.com](http://www.InternetNews.com)

**CGI Specification:** <http://hoohoo.ncsa.uiuc.edu/cgi/interface.html>

**FastCGI Web site:** [www.fastcgi.com](http://www.fastcgi.com)

**Perl Language Site:** [www.perl.com](http://www.perl.com)

**Perl Mongers:** [www.perl.org](http://www.perl.org)