

ÁNGEL COBO  
PATRICIA GÓMEZ  
DANIEL PÉREZ  
ROCÍO ROCHA

# PHP y MySQL

Tecnologías para el desarrollo de  
aplicaciones web



© Ángel Cobo, Patricia Gómez, Daniel Pérez y Rocío Rocha, 2005

«No está permitida la reproducción total o parcial de este libro, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier medio, ya sea electrónico, mecánico por fotocopia, por registro u otros métodos, sin el permiso previo y por escrito de los titulares del Copyright.»

Ediciones Díaz de Santos

Internet: <http://www.diazdesantos.es/ediciones>

E-mail: [ediciones@diazdesantos.es](mailto:ediciones@diazdesantos.es)

ISBN: 84-7978-706-6

Depósito Legal: M. 27.919-2005

Fotocomposición: P55 Servicios Culturales

Diseño de cubierta: P55 Servicios Culturales

Impresión: Fernández Ciudad

Encuadernación: Rústica-Hilo

Printed in Spain - Impreso en España

---

# Índice

<b>Agradecimientos</b> .....	<b>vii</b>
<b>Prefacio</b> .....	<b>xvii</b>
<b>1. INTERNET Y LA PROGRAMACIÓN DE ORDENADORES</b> .....	<b>1</b>
1.1. INTRODUCCIÓN .....	1
1.2. INTERNET .....	3
1.2.1. El servicio web .....	4
1.2.2. Relación cliente/servidor .....	5
1.2.3. Proceso de creación y publicación de páginas web .....	7
1.2.4. Dinamismo e interactividad en las páginas web .....	8
1.3. PROGRAMACIÓN DE ORDENADORES .....	9
1.3.1. Tipos de lenguajes de programación .....	11
1.4. PROGRAMACIÓN EN INTERNET .....	15
1.4.1. Programación del lado del cliente vs programación del lado del servidor .....	16
1.4.2. Tecnologías de programación del lado del cliente .....	18
1.4.3. Tecnologías de programación del lado del servidor .....	20
<b>2. OPEN SOURCE Y SOFTWARE LIBRE</b> .....	<b>25</b>
2.1. INTRODUCCIÓN .....	25
2.1.1. ¿Qué es <i>open source</i> ? .....	27

2.2.	BREVE RESEÑA CRONOLÓGICA .....	28
2.2.1.	Primera etapa, antecedentes .....	28
2.2.2.	Segunda etapa, desarrollo .....	28
2.2.3.	Tercera etapa, expansión .....	29
2.3.	LA CORRIENTE <i>OPEN SOURCE</i> Y SOFTWARE LIBRE, FILOSOFÍA Y CARACTERÍSTICAS .....	31
2.4.	LICENCIAS .....	35
2.4.1.	Licencias propietarias .....	36
2.4.2.	Licencias libres .....	37
2.4.3.	Licencias semilibres .....	38
2.5.	SITUACIÓN ACTUA DEL SOFTWARE <i>OPEN SOURCE</i> .....	40
2.5.1.	Software <i>open source</i> , aplicaciones para todo .....	40
2.5.2.	Comunidades <i>open source</i> .....	47
2.5.3.	Usuarios de software <i>open source</i> .....	48
2.6.	VENTAJAS, INCONVENIENTES Y PERSPECTIVAS DE FUTURO .....	51
2.6.1.	Ventajas .....	51
2.6.2.	Inconvenientes .....	52
2.6.3.	Perspectivas de futuro .....	52
<b>3.</b>	<b>CREACIÓN DE PÁGINAS WEB MEDIANTE EL LENGUAJE HTML .....</b>	<b>55</b>
3.1.	INTRODUCCIÓN .....	55
3.1.1.	Definición de HTML .....	57
3.2.	HISTORIA DE HTML .....	58
3.3.	SOFTWARE NECESARIO PARA TRABAJAR CON HTML .....	60
3.3.1.	Navegadores .....	60
3.3.2.	Editores .....	60
3.4.	ESTRUCTURA DE UNA PÁGINA WEB .....	62
3.4.1.	Complementos a la estructura básica: metatags .....	62
3.5.	CARACTERÍSTICAS GENERALES DEL LENGUAJE Y SINTAXIS DE SUS COMANDOS .....	66
3.5.1.	Caracteres especiales .....	67
3.6.	COMANDOS BÁSICOS .....	68
3.6.1.	Comandos de cabeceras para definición de títulos y secciones .....	68
3.6.2.	Comandos de cambio de estilo de texto .....	68
3.6.3.	Bloques de texto y párrafos .....	69
3.6.4.	Comandos para la generación de listas .....	71

3.6.5.	Creación de Tablas .....	76
3.6.6.	Inserción de imágenes .....	79
3.6.7.	Tratamiento del color .....	81
3.6.8.	Enlaces o hipervínculos .....	82
3.7.	MARCOS O <i>FRAMES</i> .....	85
3.7.1.	Construcción de páginas con marcos .....	85
3.7.2.	Enlaces en páginas con marcos .....	87
3.8.	FORMULARIOS .....	90
<b>4.</b>	<b>INTRODUCCIÓN A PHP .....</b>	<b>99</b>
4.1.	EL LENGUAJE PHP .....	99
4.2.	ORÍGENES Y EVOLUCIÓN DEL LENGUAJE .....	100
4.3.	PROGRAMACIÓN EN PHP: PRIMEROS EJEMPLOS .....	102
4.4.	FORMAS DE INCRUSTAR EL CÓDIGO PHP EN LOS DOCUMENTOS HTML .....	108
4.5.	ESCRITURA DEL CÓDIGO FUENTE PHP .....	111
4.5.1.	Aspectos sobre la sintaxis del lenguaje .....	111
4.5.2.	Inserción de comentarios en el código .....	112
4.6.	ELEMENTOS BÁSICOS DEL LENGUAJE .....	113
4.6.1.	Tipos de datos .....	113
4.6.2.	Variables .....	114
4.6.3.	Constantes .....	119
4.6.4.	Cadenas de caracteres y variables .....	120
4.6.5.	Operadores .....	124
<b>5.</b>	<b>PHP Y LOS FORMULARIOS DE HTML .....</b>	<b>137</b>
5.1.	FORMULARIOS DE HTML .....	137
5.2.	ENVÍO DE DATOS A PROGRAMAS PHP .....	139
5.3.	MÉTODOS DE ACCESO A LAS VARIABLES DEL FORMULARIO .....	141
5.4.	MÉTODOS DE TRANSFERENCIA DE DATOS EN FORMULARIOS .....	143
5.4.1.	Método GET .....	144
5.4.2.	Método POST .....	145
5.4.3.	Diferencias entre ambos métodos .....	146
5.5.	TRANSFERENCIA DE DATOS AL SCRIPT DESDE EL URL .....	147
5.6.	TRATAMIENTO DE FORMULARIOS CON LISTAS DE SELECCIÓN MÚLTIPLE .....	153

<b>6. SENTENCIAS DE CONTROL</b> .....	<b>157</b>
6.1. INTRODUCCIÓN .....	157
6.2. ESTRUCTURAS CONDICIONALES .....	158
6.2.1. Sentencia IF .....	158
6.2.2. La cláusula ELSE .....	162
6.2.3. Sentencia If ... ELSEIF ... ELSE .....	165
6.2.4. Sintaxis alternativa de las sentencias IF .....	167
6.2.5. Estructuras condicionales SWITCH ... CASE .....	169
6.3. ESTRUCTURAS ITERATIVAS .....	175
6.3.1. Sentencia WHILE .....	175
6.3.2. Sentencia DO ... WHILE .....	176
6.3.3. Sentencia FOR .....	179
6.3.4. Sentencia FOREACH .....	187
6.3.5. Sentencias BREAK y CONTINUE .....	188
<b>7. DEFINICIÓN DE FUNCIONES</b> .....	<b>193</b>
7.1. ¿CÓMO SE DEFINEN FUNCIONES EN PHP? .....	193
7.2. LLAMADA A LAS FUNCIONES .....	196
7.3. ARGUMENTOS DE UNA FUNCIÓN .....	199
7.3.1. Argumentos opcionales .....	199
7.3.2. Argumentos con valores por defecto .....	200
7.3.3. Listas de argumentos de longitud variable .....	201
7.3.4. Paso de argumentos por valor o por referencia .....	203
7.4. ÁMBITO DE LAS VARIABLES .....	206
7.4.1. Variables locales .....	206
7.4.2. Variables globales .....	207
7.4.3. Variables estáticas .....	208
7.5. CLÁUSULAS INCLUDE Y REQUIRE .....	209
7.6. FUNCIONES VARIABLES .....	211
<b>8. ARRAYS</b> .....	<b>215</b>
8.1. CONCEPTOS FUNDAMENTALES .....	215
8.1.1. Construcción de arrays .....	216
8.1.2. Arrays multidimensionales .....	218
8.2. RECORRIDO DE TODOS LOS ELEMENTOS DE UN ARRAY .....	219
8.3. FUNCIONES DE MANIPULACIÓN DE ARRAYS .....	225
8.3.1. Transformación de los índices .....	225
8.3.2. Subdivisión .....	226

8.3.3.	Contabilización de elementos .....	227
8.3.4.	Búsqueda de datos .....	228
8.3.5.	Generación de arrays .....	232
8.3.6.	Prolongación o truncamiento de un array .....	237
8.3.7.	Combinación de arrays .....	238
8.3.8.	Aplicación de funciones .....	240
8.3.9.	Ordenación de los elementos de un array .....	244
<b>9.</b>	<b>FUNCIONES PREDEFINIDAS .....</b>	<b>249</b>
9.1.	INTRODUCCIÓN .....	249
9.2.	FUNCIONES DE MANIPULACIÓN DE CADENAS DE CARACTERES .....	250
9.2.1.	Reconocimiento de caracteres .....	250
9.2.2.	Conversiones entre cadenas y arrays .....	250
9.2.3.	Prolongación de una cadena .....	253
9.2.4.	Modificación de una cadena .....	254
9.2.5.	Comparación de cadenas .....	256
9.2.6.	Búsqueda de datos .....	256
9.2.7.	Subdivisión de cadenas .....	258
9.2.8.	Longitud de una cadena .....	259
9.3.	FUNCIONES DE FECHA Y HORA .....	259
9.4.	FUNCIONES MATEMÁTICAS .....	267
<b>10.</b>	<b>PROGRAMACIÓN ORIENTADA A OBJETOS .....</b>	<b>269</b>
10.1.	INTRODUCCIÓN .....	269
10.2.	DEFINICIÓN DE UNA CLASE .....	272
10.2.1.	Constructores .....	275
10.3.	CONSTRUCCIÓN DE OBJETOS .....	277
10.3.1.	Acceso a las variables y métodos del objeto .....	278
10.4.	HERENCIA .....	281
10.4.1.	Definición de subclases .....	283
<b>11.</b>	<b>COOKIES .....</b>	<b>285</b>
11.1.	¿QUÉ SON LAS COOKIES? .....	285
11.2.	GENERACIÓN DE COOKIES .....	286
11.3.	RECUPERACIÓN DE LOS VALORES DE LAS COOKIES .....	288
11.4.	COOKIES DE SESIÓN Y COOKIES PERMANENTES .....	291
11.5.	VISIBILIDAD DE LAS COOKIES EN EL SITIO WEB .....	294

<b>12. MANEJO DE FICHEROS</b> .....	<b>295</b>
12.1. MECANISMOS DE ALMACENAMIENTO DE DATOS .....	295
12.2. OPERACIONES DE MANIPULACIÓN DE FICHEROS .....	296
12.2.1. Apertura y cierre de ficheros .....	296
12.2.2. Operaciones de lectura de datos .....	298
12.2.3. Lectura con formato .....	300
12.2.4. Operaciones de escritura de datos .....	303
12.2.5. Otras funciones de manipulación de ficheros .....	306
12.3. ENVÍO DE FICHEROS A TRAVÉS DE FORMULARIOS HTML ...	307
<b>13. BASES DE DATOS RELACIONALES Y EL LENGUAJE SQL</b> .....	<b>309</b>
13.1. INTRODUCCIÓN .....	309
13.2. DISEÑO DE BASES DE DATOS .....	310
13.2.1. Modelo relacional .....	310
13.2.2. Diagramas entidad/relación .....	311
13.3. UN EJEMPLO ILUSTRATIVO: CINEM@S .....	311
13.4. EL LENGUAJE SQL .....	316
13.4.1. Introducción .....	316
13.4.2. Sentencias de definición de datos .....	317
13.4.3. La sentencia INSERT .....	324
13.4.4. La sentencia DELETE .....	328
13.4.5. La sentencia UPDATE .....	329
13.4.6. La sentencia SELECT .....	330
<b>14. EL SISTEMA GESTOR DE BASES DE DATOS MYSQL</b> .....	<b>339</b>
14.1. ¿QUÉ ES MYSQL? .....	339
14.2. UTILIZACIÓN DE MYSQL .....	340
14.2.1. Arranque del servidor MySQL .....	340
14.2.2. Inicio del monitor de MySQL .....	342
14.3. EJECUCIÓN DE SENTENCIAS SQL .....	344
14.4. GESTIÓN DE USUARIOS .....	346
14.4.1. La tabla USER .....	346
14.4.2. Eliminación de usuarios .....	349
14.4.3. Establecimiento de contraseñas para los usuarios .....	350
14.4.4. Creación de nuevos usuarios .....	352
14.5. BASES DE DATOS Y TABLAS EN MYSQL .....	354
14.6. TIPOS DE DATOS .....	356
14.6.1. Tipos Numéricos .....	356



14.6.2.	Tipos cadena de caracteres .....	359
14.6.3.	Tipos enum y set .....	361
14.6.4.	Tipos fecha/hora .....	362
14.7.	INTEGRIDAD REFERENCIAL EN MYSQL .....	364
14.7.1.	Definición de claves foráneas .....	365
14.7.2.	Inserción de registros en tablas relacionadas .....	372
14.7.3.	Eliminación de registros en tablas relacionadas .....	373
14.7.4.	Actualización de registros en tablas relacionadas .....	379
14.8.	IMPORTACIÓN Y EXPORTACIÓN de datos .....	381
14.8.1.	Importación de datos .....	381
14.8.2.	Exportación de datos .....	385
<b>15.</b>	<b>PHPMYADMIN: ADMINISTRACIÓN DE BASES DE DATOS MYSQL DESDE LA WEB .....</b>	<b>389</b>
15.1.	INTRODUCCIÓN .....	389
15.2.	ENTRADA A PHPMYADMIN .....	390
15.3.	GESTIÓN DE BASES DE DATOS .....	392
15.3.1.	Consulta de las bases de datos .....	392
15.3.2.	Manipulación de las tablas .....	393
15.3.3.	Estructura de las tablas .....	398
15.4.	CREACIÓN DE NUEVAS BASES DE DATOS .....	400
15.4.1.	Creación de tablas .....	401
15.5.	OPERACIONES DE IMPORTACIÓN Y EXPORTACIÓN DE DATOS .....	403
15.5.1.	Importación de datos externos en las tablas .....	403
15.5.2.	Exportación de datos .....	405
15.6.	GESTIÓN DE USUARIOS .....	407
15.6.1.	Usuarios y privilegios .....	407
15.6.2.	Creación de usuarios .....	408
<b>16.</b>	<b>CONECTIVIDAD A BASES DE DATOS MYSQL DESDE PHP .....</b>	<b>413</b>
16.1.	INTRODUCCIÓN .....	413
16.2.	CONEXIÓN CON MYSQL DESDE PHP .....	414
16.2.1.	Apertura de la conexión .....	414
16.2.2.	Cierre de la conexión .....	415
16.3.	SELECCIÓN DE LA BASE DE DATOS .....	416
16.4.	EJECUCIÓN DE SENTENCIAS SQL SOBRE LA BASE DE DATOS SELECCIONADA .....	417

16.4.1. Inserción de datos a través de formularios.....	419
16.4.2. Recuperación de los resultados de las consultas.....	424
16.5. OTRAS FUNCIONES DE MANIPULACIÓN DE DATOS .....	450
<b>17. IMPLEMENTACIÓN DE FOROS .....</b>	<b>453</b>
17.1. INTRODUCCIÓN .....	453
17.2. ESTRUCTURA DE LA TABLA DE LA BASE DE DATOS .....	454
17.3. GENERACIÓN DE LA PÁGINA PRINCIPAL DE LOS FOROS ....	455
17.4. CONSULTA DE UN MENSAJE CON SUS RESPUESTAS .....	458
17.5. INSERCIÓN DE NUEVOS MENSAJES .....	460
<b>18. SOLUCIONES <i>OPEN SOURCE</i> BASADAS EN PHP Y MYSQL .....</b>	<b>465</b>
18.1. INTRODUCCIÓN .....	465
18.2. GESTORES DE CONTENIDO .....	466
18.2.1. Funcionamiento .....	468
18.2.2. Características .....	468
18.2.3. Ventajas e inconvenientes .....	470
18.3. GESTORES DE CONTENIDO BASADOS EN PHP Y MYSQL .....	471
18.3.1. PHP-NUKE .....	471
18.3.2. POSTNUKE .....	473
18.3.3. MAMBO SERVER.....	474
18.3.4. PHPWEBSITE .....	475
18.3.5. PHP-WCMS .....	476
18.3.6. XOOPS .....	477
18.3.7. DRUPAL.....	478
18.3.8. SITEFRAME.....	479
18.4. OTRAS SOLUCIONES <i>OPEN SOURCE</i> .....	480
18.4.1. Weblogs.....	480
18.4.2. Sistemas de foros .....	484
18.4.3. Plataformas de <i>e-learning</i> .....	486
18.4.4. Entornos de comercio electrónico .....	488
18.4.5. Sistemas de atención al cliente.....	490
18.4.6. Herramientas de gestión de proyectos.....	493
18.4.7. Otras herramientas .....	495
<b>Índice alfabético .....</b>	<b>497</b>

---

## Prefacio

El presente libro aborda una temática de plena actualidad y de gran auge en los últimos años, como es el uso de soluciones *open source* para el desarrollo de aplicaciones web. El uso conjunto del lenguaje PHP y el sistema gestor de bases de datos MySQL permite la construcción, de una manera sencilla y eficiente, de verdaderos sitios web dinámicos. En los últimos tiempos se está observando cómo son cada vez más las empresas que optan por estas tecnologías para el desarrollo de sus portales; incluso, en el caso de particulares, algunos de los servicios gratuitos de alojamiento de páginas web ofrecen la posibilidad de usar PHP en conjunción con MySQL.

Los autores de este libro somos profesores de la Universidad de Cantabria, con experiencia en la impartición de cursos sobre tecnologías de programación, desarrollo de aplicaciones web, informática de gestión y sistemas de información. Fruto de nuestra dilatada experiencia docente de estos años surge el presente libro. Parte del material que incluye ha sido usado exitosamente para la impartición de varios cursos de verano y un módulo de desarrollo de aplicaciones web en el Máster en e-Business que organiza la Universidad de Cantabria.

Nuestra intención ha sido escribir un libro que pueda ser seguido por personas con conocimientos muy básicos de lenguajes de programación y con inquietudes por el desarrollo de sitios web. En el campo de la informática, especialmente, existe un gran número de personas autodidactas y con un carácter claramente vocacional; este tipo de libros va dirigido a ellos.

Es por ello que se ha buscado un enfoque eminentemente práctico, ilustrando los diferentes conceptos con un gran número de ejemplos. De hecho, se pretende que a lo largo de los diferentes capítulos se vaya describiendo paso a paso todo el proceso de desarrollo de un sitio web verdaderamente dinámico y profesional. Para ello hemos optado por mostrar el desarrollo del sitio web de una empresa ficticia; en concreto consideraremos el sitio web de un multicine que hemos bautizado como «*Cinem@s*» y que ofrecerá la posibilidad de consultar carteleras dinámicas (mostrando siempre la información actualizada), consultar disponibilidades de localidades, registrarse como cliente,...

El libro comienza con una rápida presentación de las características generales de Internet y las diferentes tecnologías de programación que pueden ser usadas. Para pasar posteriormente a analizar la filosofía y evolución de la conocida como corriente *open source*. El tercer capítulo tiene por objeto presentar los comandos básicos del lenguaje HTML que permitan diseñar unas sencillas páginas web en torno a las que construir posteriormente la aplicación web.

Tras estos tres capítulos iniciales, los siguientes se dedican a la presentación del lenguaje PHP, su interacción con los formularios de HTML, así como a presentar las estructuras básicas presentes en todo lenguaje de programación (sentencias de control, funciones y estructuras de datos). Se ha dedicado también un capítulo a una introducción muy básica a la programación orientada a objetos; así como sendos capítulos dedicados a manejo de cookies y ficheros.

La segunda parte del libro se centra en el uso del sistema MySQL, para lo que se ha considerado oportuno introducir un capítulo previo de estudio del modelo relacional de bases de datos y el lenguaje SQL. Tras analizar las características de MySQL se muestra cómo es posible la administración remota de las bases de datos mediante una aplicación desarrollada en PHP. Una vez analizada esta aplicación de administración, los siguientes capítulos muestran la sencillez de los procesos de conectividad a MySQL desde PHP.

El libro finaliza con un capítulo en el que se presentan muy brevemente algunas de las aplicaciones *open source* más conocidas en Internet y que se basan en las tecnologías explicadas en el libro. Llegado a este punto se espera que el lector pueda no solo hacer uso de esas aplicaciones, sino también modificarlas y adaptarlas a sus necesidades particulares, aprovechando de esta forma la gran ventaja de las aplicaciones de código abierto.

Finalmente, quisiéramos mostrar nuestro agradecimiento a los Departamentos de Matemática Aplicada y Ciencias de la Computación y de Administración de Empresas de la Universidad de Cantabria por sus facilidades y apoyo para escribir

este libro. Agradecimiento que hacemos extensivo a todos nuestros alumnos durante estos años; ellos nos han servido de incentivo para embarcarnos en este proyecto editorial.

Ángel Cobo  
Patricia Gómez  
Daniel Pérez  
Rocío Rocha

Santander, febrero de 2005

# 1 INTERNET Y LA PROGRAMACIÓN DE ORDENADORES

---

---

## 1.1. INTRODUCCIÓN

La aparición a principios de los noventa del servicio web supuso una verdadera revolución en el campo de la informática y las telecomunicaciones. Con la irrupción de este nuevo servicio, Internet inició una rápida transición hacia el ámbito empresarial y supuso un enorme impulso al crecimiento de la red. Lo que había surgido en plena *guerra fría* como un proyecto militar y que posteriormente fue dirigiéndose hacia el ámbito científico y académico, se empezó a convertir en un perfecto “escaparate virtual” en el que las empresas pudieran ofrecer sus productos y servicios rompiendo barreras geográficas y de comunicaciones. En la actualidad las empresas no ven únicamente el servicio web como un mero escaparate o medio publicitario de enorme difusión. Internet, y en particular el servicio web, abre a las empresas enormes posibilidades. La utilización de tecnologías web permite agilizar los procesos, mejorar la productividad y aumentar la eficacia, además de abrir las puertas a nuevas formas de negocio en el mercado global que facilita Internet (*e-business*).

Por supuesto, no han sido las empresas las únicas beneficiadas con el desarrollo de Internet y del servicio web. Las instituciones públicas tienen también nuevas formas de ofrecer servicios a los ciudadanos (*e-gouvernement*), los usuarios individuales tienen nuevas formas de adquirir productos (*e-commerce*) o nuevas formas de formarse e instruirse (*e-learning*),...

Para poder realmente obtener todos estos beneficios ha sido preciso desarrollar nuevas tecnologías que consigan hacer del servicio web un servicio dinámico e interactivo. En sus orígenes el servicio World Wide Web fue concebido como un sistema flexible de compartir información multimedia entre equipos heterogéneos a través de redes informáticas. Para ello fue desarrollado un sistema de generación de documentos a través de un lenguaje estándar: el lenguaje HTML. Los documentos generados de esta forma podían incorporar texto y elementos gráficos, pero eran documentos totalmente estáticos. Posteriormente fueron desarrollándose diferentes tecnologías que, trabajando en conjunción con el lenguaje HTML, pudieran paliar estas carencias. Así por ejemplo, Netscape incorporó en la versión 2.0 de su célebre navegador un intérprete de un nuevo lenguaje que podía ser intercalado entre el código HTML y que permitía realizar operaciones no disponibles en un lenguaje puramente descriptivo como es HTML; dicho lenguaje, llamado en sus orígenes LiveScript, adquirió posteriormente el nombre de JavaScript. Unos años más tarde la empresa Sun Microsystems lanzó un revolucionario lenguaje de programación, el lenguaje Java, que permitía incrustar en las páginas web programas con las prestaciones propias de cualquier lenguaje de programación. Microsoft por su parte también se unió a esta evolución primero con sus lenguajes de script: VBScript basado en Visual Basic y JScript, versión Microsoft de JavaScript, y, más recientemente, con las tecnologías .NET. Igualmente se han venido desarrollando diferentes extensiones del propio lenguaje HTML con objeto de aumentar su dinamismo; en este contexto se podría citar el HTML dinámico (DHTML) y el lenguaje XML.

Internet ha jugado un papel esencial en la expansión de la denominada corriente *open source*. Los defensores de esta corriente defienden el desarrollo de aplicaciones informáticas y su distribución libre, de forma gratuita; pero no solo eso sino que también ponen a disposición de los usuarios el código fuente de los programas desarrollados. Se trata, en definitiva, de que los usuarios puedan utilizar los programas sin ninguna restricción y puedan conocer si lo desean su funcionamiento interno. El ejemplo más emblemático de esta corriente *open source* lo constituye el sistema operativo Linux. En los últimos años esta corriente se ha desarrollado enormemente y ya se pueden obtener en la propia red Internet todo tipo de programas basados en esta filosofía: servidores y navegadores web, entornos de programación, editores, herramientas ofimáticas,... Grandes empresas como IBM o Yahoo, por ejemplo, han apostado muy fuerte por las soluciones *open*

*source* y, sin lugar a dudas, el impacto de este tipo de soluciones parece destinado a incrementarse en el futuro debido a sus altas prestaciones y calidad, su menor coste y su alta difusión.

Las dos herramientas que se analizarán en este libro surgen de esta corriente. Por un lado el lenguaje PHP, que nació como un lenguaje para realizar un seguimiento de visitas de páginas personales, se ha convertido en uno de los referentes actuales en los denominados lenguajes de script. Por otro lado, el sistema gestor de bases de datos MySQL se presenta como una herramienta con altas prestaciones para el desarrollo de bases de datos, especialmente apropiado para ser usada por pequeñas organizaciones o empresas. La utilización conjunta de ambos: PHP y MySQL permite llegar a desarrollar interesantes aplicaciones web que puedan cubrir las necesidades de pequeñas empresas que quieran fortalecer su presencia en Internet o usuarios individuales que quieran generar verdaderas páginas dinámicas. Como complemento a ambas herramientas, en Internet se pueden encontrar igualmente gestores de contenidos, aplicaciones basadas en PHP que permiten a los usuarios finales aprovechar las ventajas de estas tecnologías sin necesidad de tener grandes conocimientos sobre su funcionamiento y sintaxis de los lenguajes. En el presente libro se darán referencias de algunas de estas herramientas de gestión de contenidos y otras herramientas para la creación de aplicaciones web que han sido desarrolladas igualmente haciendo uso de las dos tecnologías que se analizarán en este libro: el lenguaje PHP y el gestor de bases de datos MySQL. El Capítulo 18 presentará algunas de estas herramientas.

## **1.2. INTERNET**

El objeto de esta sección no es tanto definir lo que es Internet como el poner de manifiesto algunas de las características más destacadas de la misma que influyen decisivamente en las tecnologías de desarrollo vinculadas al servicio web, como es el caso del lenguaje PHP. Se asume que cualquier lector interesado en este libro es usuario de Internet y conoce los diferentes servicios que la red ofrece, posiblemente también su interés en el desarrollo de aplicaciones web es el que le ha impulsado a adquirir este libro.

Internet (INTERconnected NETwork) es una red de redes de ordenadores de todo tipo que se comunican mediante un lenguaje común: el conocido como protocolo TCP/IP. Esa primera característica de la heterogeneidad de los equipos conectados es clave para entender el funcionamiento de todos los servicios de la red y para comprender la necesidad de la portabilidad en cualquier desarrollo que se quiera hacer en el ámbito de Internet. A nivel de programadores de aplicaciones web, que



es en el que se sitúa el presente libro, no es necesario disponer de conocimientos técnicos sobre los protocolos de comunicación en los que se basa Internet.

Los desarrollos de aplicaciones realizados en Internet tienen también un campo de aplicación en aquellas redes privadas que usan los mismos sistemas y protocolos que Internet: las denominadas Intranets.

Otro de los aspectos a tener en cuenta es el carácter distribuido de la red. Ese carácter totalmente distribuido se concreta a todos los niveles: en el aspecto geográfico no existe ningún nodo central de la red, de hecho, los orígenes de Internet se sitúan en un proyecto militar del Gobierno de los Estados Unidos para crear una red que no fuera vulnerable ante el ataque a alguno de sus nodos. En el aspecto económico, tampoco existe ningún gobierno o institución que mantenga la red sino que son las propias subredes que la componen las encargadas de su propio mantenimiento. El carácter distribuido también se manifiesta en el aspecto político ante la ausencia de un gobierno central de la red; lo que sí existen son diversas organizaciones o asociaciones que tratan de establecer diferentes estándares para el desarrollo de la red.

Aunque el servicio web es actualmente el servicio más conocido y utilizado de la red Internet, conviene recordar que no es el único. Los tres servicios originarios de la red: correo electrónico (e-mail), transferencia de fichero (FTP) y acceso remoto (Telnet), siguen estando presentes y siguen siendo ampliamente utilizados. Pero podrían citarse otros servicios, algunos muy conocidos y otros en cierta decadencia al haber absorbido el propio servicio web sus funciones: servicios de noticias (news), gopher, servicios de búsqueda de archivos (Archie), Verónica, servicios de localización,... En los últimos años también están teniendo mucho auge las aplicaciones P2P.

### **1.2.1. EL SERVICIO WEB**

El servicio WWW, o simplemente Web, se podría definir como un amplio sistema multimedia de acceso a información heterogénea distribuida por toda la red en forma de documentos hipertextuales (hipertextos). Como ya fue comentado en la introducción de este capítulo, este servicio surgió en 1990 en el CERN (Centre Européen de Recherche Nucléaire) con el objetivo de facilitar la distribución de información entre equipos investigadores geográficamente dispersos. Se buscaba que los recursos disponibles en formato electrónico fuesen accesibles para cada investigador desde su propia terminal de forma clara y simple, posibilitando el salto entre elementos de información conexos. En definitiva, se trataba de integrar todos los recursos existentes en una red hipertextual. Aunque el nacimiento del

servicio se sitúa en 1990, es en 1991 cuando el sistema desarrollado en el CERN se abre a Internet, apareciendo en 1992 el primer navegador web: Mosaic.

El término hipertexto que empezó a hacerse popular a partir de la aparición de este servicio tiene, sin embargo, su definición en un trabajo de Ted Nelson en 1965, la definición original del término es:

*"Un cuerpo de material escrito o gráfico interconectado de un modo complejo que no se puede representar convenientemente sobre el papel; puede contener anotaciones, adiciones y notas de los estudiosos que lo examinan".*

En una definición más moderna y aplicable al concepto de hipertexto en Internet, se podría decir que un hipertexto es un documento multimedia, es decir, integrando bajo una plataforma informática todas las tecnologías de la información, y que incorpora relaciones estructurales que enlazan el documento con otros documentos o recursos.

Algunas de las características destacadas de los hipertextos son:

- Almacenamiento de un gran volumen de información.
- Facilidad de acceso y consulta.
- Presentación de una forma más agradable.
- Uso de todas las tecnologías de la información.
- Permiten una “navegación” individualizada.
- Estructuración multidimensional.
- Multiplataforma.
- Dinamismo e interactividad.

## **1.2.2. RELACIÓN CLIENTE/SERVIDOR**

Todos los servicios que ofrece Internet, y por supuesto entre ellos el servicio web, se basan en la denominada relación cliente/servidor. El comprender bien esta relación es esencial para entender el funcionamiento posterior de lenguajes como PHP. En Internet se pueden encontrar dos tipos de equipos conectados:

- *Servidores*: ordenadores que ofrecen sus servicios al resto de equipos conectados. Suelen tener una presencia estable en la red, lo que se concreta en tener asignadas direcciones IP permanentes. En ellos es donde están alojadas, por ejemplo, las páginas web.

- *Clientes*: equipos que los usuarios individuales utilizan para conectarse a la red y solicitar servicios a los servidores. Durante el tiempo de conexión tienen presencia física en la red. Normalmente los proveedores de acceso a Internet asignan a estos equipos una dirección IP durante su conexión, pero esa dirección es variable, es decir, cambia de unas conexiones a otras (IP dinámica).

Los conceptos de cliente y servidor se suelen utilizar con dos significados diferentes, en referencia al hardware el sentido es el indicado anteriormente, el servidor hace referencia al equipo remoto al que se realiza la conexión y el cliente sería el equipo local utilizado para efectuar dicha conexión. Pero también se utilizan esos conceptos en referencia al software:

- *Programa servidor* es el programa que debe estar ejecutándose en el equipo servidor para que este pueda ofrecer su servicio. Un documento HTML sin más almacenado en el equipo remoto no basta para que sea accesible como página web por el resto de usuarios de Internet, en ese equipo debe estar ejecutándose una aplicación servidor web. Uno de los programas servidores web más conocido y utilizado es Apache, programa que también pertenece a la corriente *open source*. Existen otros servidores web como el Personal Web Server (PWS) o el IIS disponibles en los equipos Windows. En el caso de otros servicios como el correo electrónico o la transferencia de ficheros se necesitarían igualmente los correspondientes programas en el servidor.
- *Programa cliente* es en este caso el software necesario en el equipo cliente para tener acceso al correspondiente servicio. Así por ejemplo, los navegadores como el Internet Explorer o Mozilla son ejemplos de clientes web; un programa como Outlook es un ejemplo de cliente de correo electrónico y programas como WS\_FTP o CuteFTP son ejemplos de clientes FTP.



**Figura 1.1** *Relación cliente/servidor. Los clientes realizan peticiones de servicio a los servidores*

### 1.2.3. PROCESO DE CREACIÓN Y PUBLICACIÓN DE PÁGINAS WEB

El proceso de creación y publicación de una página web debe pasar por una serie de fases:

- *Definición de la página:* toda página web tiene detrás un código fuente que la define. Cuando se accede a una página web, aunque en el monitor se puedan visualizar documentos con imágenes y texto de diferentes tamaños, colores y formatos, debe tenerse presente que detrás de eso hay un documento de texto sin ningún tipo de formato y que incorpora una serie de instrucciones o comandos que son los que realmente generan la página que se visualiza. Ese documento fuente está definido en un lenguaje especial: el conocido HTML (HyperText Markup Language). Se trata de un lenguaje puramente descriptivo que incorpora una serie de comandos o etiquetas (*tags*) que permiten definir la estructura lógica del documento, dar formato al texto, añadir elementos no textuales,... Se asume que los lectores interesados en este libro tienen un cierto conocimiento de este lenguaje de definición de hipertextos, no obstante, se ha considerado oportuno incluir un capítulo introductorio en el que se presenten los comandos básicos de HTML (Capítulo 3). Para aquellos lectores sin conocimientos de HTML, una comprensión rápida de los comandos expuestos en dicho capítulo es más que suficiente para poder entender el resto del libro.
- *Publicación del documento:* una vez creado el documento HTML que define la página web, el siguiente paso es evidentemente publicarla para que esté disponible para el resto de usuarios de Internet. La publicación implica la transferencia del documento a un equipo servidor que disponga de un programa de servidor web. Puede optarse por utilizar algún servidor de alojamiento gratuito de páginas o utilizar los espacios que los proveedores de acceso a Internet suelen ofrecer a sus clientes. Otra posibilidad sería configurar un equipo propio para que actúe de servidor, para ello sería preciso contratar con algún proveedor una dirección IP fija y registrar el dominio que se quiera utilizar.
- *Acceso a las páginas web:* en el momento en que una página es publicada en el servidor, cualquier usuario de Internet podría acceder a ella. Para ello es preciso, por un lado que el usuario utilice un programa adecuado (el navegador, explorador o cliente web), y que el código que define la página sea transferido por la red utilizando el protocolo *http* (hypertext transfer protocol). Sobre este último aspecto, en principio la mayoría de usuarios no deberían preocuparse. El navegador web es el que se encarga de interpretar los

comandos HTML que se reciben y producir a partir de ellos la página web. Existen diferentes navegadores web que se pueden utilizar, por ejemplo, Internet Explorer, Netscape Navigator, HotJava, Mozilla,... este último sigue la corriente *open source*.

#### **1.2.4. DINAMISMO E INTERACTIVIDAD EN LAS PÁGINAS WEB**

HTML es un lenguaje puramente descriptivo que permite definir las páginas web pero que en modo alguno se puede considerar un lenguaje de programación. Con HTML no se pueden generar estructuras iterativas o condicionales, no se pueden definir funciones que sean utilizadas en distintos puntos del documento, no se pueden declarar variables, no se pueden realizar cálculos matemáticos,... Las páginas creadas únicamente con HTML son básicamente estáticas, es decir, siempre muestran la misma información y no ofrecen ningún grado de interactividad con el usuario. Los únicos elementos de HTML que podrían de alguna forma considerarse interactivos son los formularios a través de los cuales se solicita información al usuario.

Si se requiere aumentar el dinamismo e interactividad de las páginas se hace por tanto obligado el recurrir a otros lenguajes y tecnologías como las que se abordan en este libro. Esas dos características: dinamismo e interactividad son los dos elementos clave que se deben tratar de potenciar para desarrollar verdaderas aplicaciones web.

Pero, ¿qué es una página dinámica? Piénsese, por ejemplo, en un multicine que quiere publicar en Internet la información actualizada sobre horarios y películas que se proyectan en cada una de sus salas. Evidentemente, sería muy sencillo generar una simple página en HTML con una tabla en la que se muestre esa información; incluso sin necesidad de tener ningún tipo de conocimiento sobre HTML, por ejemplo, se podría escribir en Word y usar la opción de "*Guardar como página web...*". Hasta aquí sencillo, pero la información en sí es un elemento dinámico, las películas que se proyectan en cada sala cambian y por tanto la página web debería ser actualizada. ¿Tiene sentido tener que modificar el documento HTML cada vez que se produzca un cambio en la cartelera?, ¿no sería muchísimo más rentable disponer de un sistema que modifique de forma automática la información que muestra la página web? Esto es dinamismo, y esto no se puede conseguir solo con HTML. A lo largo de los diferentes capítulos de este libro el usuario aprenderá cómo resulta muy sencillo generar una base de datos con MySQL con toda la información de las películas a proyectar y una página web con un programa PHP que se encargue, cada vez que un usuario solicita la página, de hacer una consulta a la base de datos para obtener la cartelera actualizada y generar con ella de forma automática el código HTML que se envía al usuario. Este mismo ejemplo se irá desarrollando paso a paso en los próximos capítulos y servirá de hilo conductor del resto del libro.

Otro ejemplo de dinamismo puede ser el preprocesamiento de datos introducidos en formularios, comprobando por ejemplo la validez sintáctica de direcciones de correo electrónico o la pertenencia a determinados rangos de valores numéricos.

En lo referente a interactividad, se trata de permitir que entre el usuario que acceda a la página y la propia página exista un cierto grado de comunicación que no se limite a que la página muestre la información al usuario. El usuario, por ejemplo, le podría solicitar a la página que realice algún cálculo. Volviendo al ejemplo de los multicines, supongamos que se desea que el usuario pueda hacer un cálculo de forma automática de los importes de las entradas; mediante un formulario el usuario puede introducir el número de entradas a adquirir, la sesión para la que se desean y determinados parámetros que puedan afectar al precio (ser estudiante, disponer de un vale de descuento,...) y la página debe calcular de forma automática el importe total a pagar. Esto es interactividad.

### **1.3. PROGRAMACIÓN DE ORDENADORES**

La programación de ordenadores se podría definir como el conjunto de técnicas, métodos y reglas para poder construir programas de ordenador legibles, correctos y eficientes. Un programa de ordenador no es más que una secuencia de instrucciones en las que se le indican a la máquina las órdenes o acciones a realizar; se podría entender por tanto la programación como el arte de decir a una máquina lo que queremos que haga de una manera que pueda entenderlo.

Como todos los campos de la informática, la programación de ordenadores ha sufrido una importante evolución en las últimas décadas. Los orígenes se sitúan en la década de los 50 con la aparición de los primeros lenguajes de programación: Fortran (1954), Cobol (1954) y Algol (1957). Estos primeros lenguajes estaban muy orientados hacia aplicaciones concretas; así por ejemplo, Fortran es un lenguaje orientado hacia el cálculo científico mientras que Algol lo es hacia aplicaciones de gestión. Los años 60 vienen marcados por el nacimiento de la programación estructurada y la aparición de lenguajes como PL/1 de IBM (1960) o el BASIC (1963).

Es en los años 70 cuando aparecen dos de los lenguajes estructurados de mayor difusión: el Pascal (1970) y el C (1972); este último evolucionaría posteriormente hacia el C++ y hoy en día es el lenguaje más utilizado en el desarrollo de aplicaciones. La programación estructurada se basa en el desarrollo de módulos o funciones independientes que puedan ser utilizadas en cualquier momento. Es destacable también en esta década de los 70 el desarrollo del modelo relacional de

bases de datos, modelo en el que se basan la mayoría de los sistemas gestores de bases de datos actuales, entre ellos MySQL.

Los 80 vienen marcados por la aparición del ordenador personal y el nacimiento de la microinformática. La informática deja de ser algo exclusivo de las grandes empresas e instituciones y se acerca al público en general. En esta época aparecen también los primeros sistemas operativos con interfaces de usuario gráficas: el célebre “Mac” de Apple Macintosh. Años más tarde Microsoft seguiría esos pasos con su sistema operativo Windows. La aparición de este tipo de interfaces también afectaría notablemente a la evolución de los lenguajes de programación, en concreto surge la necesidad de desarrollo de técnicas de programación basadas en eventos. En los años 80 aparece también una nueva forma de entender la programación: la programación orientada a objetos, así surgen los lenguajes ADA (1980) y C++ (1985). La base en la que se apoya este estilo de programación es el entender los objetos como entidades compuestas de acciones y datos y, por tanto, a la hora de programar las funciones (acciones) y las variables (datos) deben entenderse como componentes de una misma unidad.

La década de los 90 se inicia con la aparición del revolucionario servicio web y la necesidad cada vez más patente de orientar los desarrollos hacia la integración de aplicaciones. El servicio web, como ya se comentó, rápidamente crea la necesidad de nuevas formas de programación y así en 1995 nace el lenguaje Java, primer lenguaje pensado para integrar directamente programas en las páginas web. Los programas Java que se integran en esas páginas se denominan *applets*.

En la actualidad, la evolución en los diferentes sistemas operativos, de los nuevos servicios de Internet y del propio hardware influyen en las prestaciones recomendadas que debe tener un lenguaje de programación, algunas de las que se podrían citar son:

- *Programación estructurada*: en la actualidad todos los lenguajes de programación soportan este tipo de programación, en la que las instrucciones se agrupan en bloques constituyendo módulos que se llaman unos a otros.
- *Programación orientada a objetos*: los lenguajes de programación de mayor éxito se basan en la construcción de clases de objetos. Cada clase engloba, por un lado las acciones que pueden ser realizadas con esos objetos y, por otro, los datos o características asociadas a los mismos. Algunos lenguajes actuales, aunque no pueden ser considerados orientados a objetos, sí que incorporan algunas características propias de estos lenguajes, trabajando también con objetos.

- Programación guiada por eventos: los programas deben ser capaces de dar respuesta a las diferentes acciones que el usuario efectúa sobre la interfaz del programa. Por ejemplo, cuando un usuario elige una opción de menú o pulsa un botón se genera un evento que debe ser detectado, identificado y tratado por el programa.
- *Programación concurrente*: la aparición cada vez con más frecuencia de equipos con más de un procesador plantea nuevas formas de programación; un programa podría ser susceptible de dividirse en varias tareas que puedan estar realizándose de forma simultánea, cada tarea puede ser ejecutada por un procesador diferente o bien utilizar mecanismos de reparto del tiempo de procesador. Esta es la base de la programación paralela o concurrente.
- *Prestaciones multimedia*: los programas deben ser capaces de manejar todo tipo de información y recursos; deben estar preparados para trabajar con elementos gráficos en diferentes formatos, animaciones o vídeo, sonido,...
- *Portabilidad*: evidentemente los programas son desarrollados para que puedan ser utilizados por diferentes usuarios y no siempre está garantizado que los usuarios tengan equipos similares. La portabilidad de los programas permite que estos puedan ser ejecutados sobre diferentes plataformas informáticas. Sin embargo, la portabilidad puede entenderse a dos niveles: a nivel de código fuente la portabilidad implica que el código puede ser compilado sobre las dos plataformas para obtener dos versiones diferentes del programa. En algunos casos, la portabilidad se consigue a nivel no de código fuente sino del código resultado del proceso de compilación; esto es lo que ocurre por ejemplo con el lenguaje Java. En Internet, la portabilidad es un concepto clave por cuanto a la red están conectados todo tipo de equipos.
- *Integración de aplicaciones*: cada vez es más necesario que los lenguajes de programación incorporen mecanismos sencillos para conectarse con otras aplicaciones. Quizás el caso más claro de ello sea la conectividad a bases de datos; los programas deben ser capaces de establecer esa conexión y realizar consultas sobre los datos almacenados.

### **1.3.1. TIPOS DE LENGUAJES DE PROGRAMACIÓN**

Los lenguajes de programación pueden ser clasificados de acuerdo a varios criterios. Una de las primeras clasificaciones que se suele efectuar es la distinción entre lenguajes de bajo nivel y de alto nivel. La programación en los primeros resulta más dificultosa puesto que las instrucciones están muy próximas al



hardware del equipo y resultan difíciles de entender por un programador no especialista. El ejemplo clásico de lenguaje de bajo nivel es el lenguaje ensamblador.

La mayor parte de los programadores optan por utilizar lenguajes cuyo código resulta más fácil de entender, por cuanto sus reglas sintácticas se asemejan más a la forma de comunicarse las personas; son lenguajes que están “más cerca” del programador pero más lejos de la máquina a la que van dirigidos. Estos lenguajes son los denominados "lenguajes de alto nivel" y a ellos pertenecen los lenguajes de programación más conocidos.

Cuando se está desarrollando un programa usando un lenguaje de programación se genera un código (código fuente) que es comprensible para todo aquel usuario que tenga los conocimientos suficientes sobre el correspondiente lenguaje, pero que en ningún caso es comprensible directamente para la máquina. Los ordenadores trabajan internamente mediante circuitos electrónicos que admiten dos posiciones: abierto o cerrado (1 ó 0) y por tanto, toda orden a dar a la máquina debe ser planteada en última instancia como secuencias de ceros y unos (código binario). Parece claro por tanto que se necesita un proceso de “traducción” del código fuente que nosotros entendemos a instrucciones entendibles por la máquina. Ese proceso de traducción se puede realizar de dos maneras, y eso da pie a establecer una nueva clasificación de los lenguajes de programación:

- *Lenguajes compilados*: en un lenguaje compilado el código fuente pasa por un proceso denominado "compilación" en el que se genera un código denominado "objeto", que una vez enlazado con otros posibles módulos de código objeto necesarios, genera el fichero ejecutable con el programa. Ese fichero ejecutable es lo único necesario para poder utilizar el programa y contiene todas las instrucciones del mismo pero en el formato entendible por la máquina. El aspecto más importante a destacar es que el proceso de compilación se realiza con anterioridad a cualquier ejecución o uso del programa; en ese proceso se comprueba la validez sintáctica del programa y si todo es correcto se genera el ejecutable. Si se produce un error en la compilación el programa no podrá ser utilizado.
- *Lenguajes interpretados*: en los lenguajes interpretados, la traducción de las instrucciones se va realizando de forma secuencial por una aplicación, denominada "intérprete", al mismo tiempo que se ejecuta el programa. De esta forma, si llegado un punto del programa el intérprete se encuentra con una instrucción errónea, el programa no continúa pero sí que habrá podido ejecutar todas las sentencias previas. A diferencia de los lenguajes

compilados, la verificación de la corrección sintáctica del programa no se realiza antes de la ejecución sino al mismo tiempo.

Por poner un símil que sirva para clarificar la diferencia entre ambos tipos de programas, supóngase que una persona recibe una lista con una serie de tareas a realizar. Podría optar por utilizar dos estrategias:

- Antes de comenzar la persona pierde unos minutos leyendo toda la lista y analizando la coherencia o factibilidad de las tareas que se le están encomendando. Si todas las tareas son coherentes realiza un proceso de planificación para determinar la forma de realizarlas todas. Pero si detecta alguna tarea que no entiende o le resulta incoherente, informa a la persona que le ha encargado las tareas y opta por no iniciar sus labores a la espera de una posible rectificación en el listado de tareas (enfoque de un programa compilado)
- Nada más recibir la lista de tareas, la persona lee la primera labor encomendada y la realiza. A continuación pasa a la segunda, la lee y la realiza, y así sucesivamente. Si todo ha ido bien, habrá realizado todas las tareas, pero si al llegar a un punto de la lista se encuentra con una tarea errónea o no coherente, opta por abandonar el resto del trabajo e informar a la persona que le encargó las tareas de tal extremo (enfoque de un programa interpretado)



**Figura 1.2** Necesidad de un proceso de traducción del código fuente para que el ordenador pueda entenderlo. Dicha traducción puede realizarse mediante un proceso de compilación o mediante el uso de un intérprete

A primera vista, se podría pensar que un lenguaje compilado es más seguro en el sentido de no producir errores a la hora de ejecutar el programa. En principio, si se producen errores sintácticos en el programa, estos son detectados en la compilación y por tanto el programa nunca se ejecutaría hasta que estos errores fuesen corregidos. Efectivamente, esto es así, pero tampoco se puede descartar que, a pesar de no haber errores sintácticos en el código fuente, se produzcan errores a la hora de ejecutar el programa. Piénsese en un programa que tiene una instrucción en la que se deben dividir dos números almacenados en dos variables y guardar el resultado en una tercera variable; la instrucción puede ser sintácticamente correcta, pero ¿qué pasa si a la hora de ejecutar el programa el segundo de los números es cero?, se producirá un error de tipo aritmético debido a la división por cero y si el programa no está preparado para ello, se abortará bruscamente su ejecución. Este tipo de errores son los que se denominan errores en tiempo de ejecución.

En un lenguaje interpretado, todos los errores son detectados en tiempo de ejecución, tanto los debidos a errores sintácticos como los debidos a condiciones singulares producidas a la hora de ejecutar el programa.

Desde un punto de vista práctico los lenguajes compilados resultan más poderosos pero los interpretados resultan más flexibles. El proceso de compilación genera códigos ejecutables fuertemente dependientes de la máquina a la que van dirigidos, con lo que se pierde la portabilidad del programa final aunque sí que se puede en algunos casos tener portabilidad del código fuente. Por ejemplo, si se ha desarrollado un programa en lenguaje C siguiendo el estándar ANSI, ese código puede ser compilado por un compilador de C en Windows para generar la versión ejecutable en Windows del programa, y posteriormente repetir la compilación en Unix con un compilador diferente para obtener la versión Unix. Por supuesto, la portabilidad del código solo es posible si no se hace uso en el programa de aspectos particulares de una plataforma concreta.

Con un lenguaje interpretado se facilita la portabilidad, ya que lo único necesario es disponer en cada plataforma del intérprete adecuado. Esta es la razón por la cual muchos de los lenguajes para el desarrollo de aplicaciones web son interpretados, ya que en Internet la portabilidad es imprescindible. Ejemplos de lenguajes interpretados son JavaScript, VBScript y el propio PHP.

Otra de las ventajas de los lenguajes interpretados puede ser la mayor facilidad para su aprendizaje y la simplificación en el proceso de desarrollo de las aplicaciones. En el caso de los lenguajes interpretados para el desarrollo de aplicaciones web, no se requiere además ningún tipo de herramienta de desarrollo como puedan ser compiladores; los intérpretes van integrados en los navegadores web en un caso y en los servidores web en otro. El código fuente además se

incrusta dentro del código HTML con lo que incluso no sería necesario ningún tipo de editor específico.

Mención especial merece el lenguaje Java: se dice de él que es un lenguaje que combina el poder de los lenguajes compilados y la flexibilidad de los interpretados. Cuando se quiere integrar en una página web un programa Java (applet Java), se necesita un proceso de compilación que genera un código binario a partir del código fuente, pero este código no es ejecutable directamente sino que se ejecuta por un intérprete que incorpora el navegador web que el usuario utilice al acceder a la página. Este pequeño artificio es el que permite conseguir la portabilidad del código compilado Java.

Los lenguajes interpretados que se utilizan en Internet para aumentar las prestaciones de las páginas web se suelen denominar lenguajes de script y a los programas con ellos desarrollados se les denomina scripts o guiones. Como ya se ha comentado, el lenguaje PHP pertenece a esta categoría al igual que los otros dos lenguajes de script más conocidos: JavaScript y VBScript.

## **1.4. PROGRAMACIÓN EN INTERNET**

Como ya ha quedado de manifiesto, para poder hacer uso de toda la potencialidad del servicio web cada vez más se requiere la utilización de lenguajes de programación que complementen al lenguaje HTML. A la hora de decidir qué tecnología o lenguaje concreto se puede utilizar para el desarrollo de una aplicación web deben plantearse algunas preguntas cuya respuesta puede condicionar la elección final:

- *¿Cuándo se realizarán las acciones?* Por ejemplo, se puede querer que el programa sea ejecutado al cargar la página que lo integra o, por el contrario, no ejecutar el programa hasta que se produzca determinado evento sobre la página (mover el ratón, pulsar un botón, situar el curso sobre algún elemento de la página,...). En este segundo caso se requiere un lenguaje que admita programación guiada por eventos, y la elección de un lenguaje como JavaScript, por ejemplo, sería más adecuada que la elección de PHP.
- *¿Cuál será el formato del conjunto de ordenes?* Se puede optar por generar programas de forma independiente que se integren en la página después de un proceso de compilación, tal como ocurre en el caso de Java, o por el contrario se puede desarrollar la aplicación en base a scripts o guiones que, sin necesidad de compilación, sean incrustados directamente entre el código HTML, tal como se hace en PHP, JavaScript o VBScript.

- *¿Quién ejecutará o interpretará las ordenes?* Como se ha comentado, en el servicio web intervienen dos equipos, el servidor en el que está alojada la página y el cliente que recibe el código HTML y genera la página en sí. Cuando existe también un programa o script integrado en la página, la duda es cuál de los dos equipos es el encargado de ejecutar las órdenes. En PHP, los programas son ejecutados por el servidor mientras que en JavaScript es el cliente el que los ejecuta.

### 1.4.1. PROGRAMACIÓN DEL LADO DEL CLIENTE VS PROGRAMACIÓN DEL LADO DEL SERVIDOR

La respuesta a la última pregunta planteada en la sección anterior da pie a la clasificación de las tecnologías de programación en Internet en dos categorías:

- *Programación del lado del cliente:* los programas residen junto a la página web en el servidor pero son transferidos al cliente para que este los ejecute. Java, JavaScript, VBScript son lenguajes de programación del lado del cliente.
- *Programación del lado del servidor:* los programas son ejecutados por el servidor y lo que se envía al cliente es la respuesta o resultado de dicha ejecución. Lenguajes como PHP o Perl pertenecen a esta categoría.



Figura 1.3 Diferencia entre la programación del lado del servidor y la del lado del cliente

Cada una de estas estrategias tiene evidentemente sus ventajas y sus inconvenientes, en cualquier caso no son excluyentes, ya que en una misma página pueden incorporarse por ejemplo scripts en PHP para ser ejecutados por el servidor y scripts en JavaScript para ser ejecutados por el cliente. En definitiva, se trata de aprovechar las ventajas de cada tecnología en el desarrollo de las aplicaciones web.

<b><i>Programación del lado del cliente</i></b>	<b><i>Programación del lado del servidor</i></b>
– <i>Los programas residen en el servidor pero se ejecutan en el cliente</i>	– <i>Los programas residen y son ejecutados por el servidor</i>
– <i>Se descarga de trabajo a los servidores</i>	– <i>El trabajo recae sobre los servidores pudiendo llegar a sobrecargarse</i>
– <i>La ejecución del programa requiere una transmisión por la red del código necesario para ello</i>	– <i>Al cliente solo se le transfiere el resultado de la ejecución del programa</i>
– <i>Las respuestas a las acciones de los usuarios sobre el programa pueden ser invocadas sin necesidad de realizar transmisiones por la red</i>	– <i>Una vez enviada al usuario la respuesta del programa, cualquier petición adicional del cliente requiere una nueva conexión con el servidor y la ejecución en él de un nuevo programa</i>
– <i>Para la correcta ejecución del programa se requiere que el cliente tenga instalados programas o plug-ins adecuados</i>	– <i>En los equipos de los clientes no se necesita ningún software especial, todo lo necesario debe estar instalado en el servidor</i>
– <i>Si en un cliente no está instalado alguno de los programas intérpretes o plug-ins, la página no se ejecutará correctamente</i>	– <i>Todos los clientes podrán visualizar correctamente la página</i>
– <i>Al transferirse el código, el cliente tiene acceso a dicho código y puede obtener a partir de él información que pueda resultar comprometida</i>	– <i>El código fuente permanece en el servidor, se conserva su privacidad y los clientes no tienen acceso a él</i>
– <i>Se pueden integrar los programas en las páginas alojadas en cualquier servidor web</i>	– <i>La mayoría de los servicios de alojamiento gratuito de páginas no admiten este tipo de programación</i>

**Tabla 1.1** *Diferencias entre la programación del lado del cliente y del lado del servidor*

## **1.4.2. TECNOLOGÍAS DE PROGRAMACIÓN DEL LADO DEL CLIENTE**

En esta sección se presentan algunas de las tecnologías de programación del lado del cliente más conocidas y utilizadas. Se trata únicamente de conocer sus principales características para compararlas posteriormente con las del lenguaje PHP.

### **JavaScript**

JavaScript es un lenguaje interpretado basado en guiones que son integrados directamente en el código HTML. El código es transferido al cliente para que este lo interprete al cargar la página. Con JavaScript no pueden crearse programas independientes.

La primera versión de este lenguaje apareció con el navegador Netscape 2.0 en 1995, con el nombre original de LiveScript y soportando gran cantidad de las instrucciones que tiene en la actualidad. La versión JavaScript 1.1 se diseñó con la llegada de las versiones 3.0 de los navegadores e incorporó algunas funcionalidades nuevas como el tratamiento dinámico de imágenes y la creación de arrays. Es esta versión la primera que se incorpora al explorador de Microsoft. En los navegadores 4.0 de Microsoft y Netscape se incorporó ya un intérprete para una nueva versión del lenguaje, el JavaScript 1.2. Con esta versión se inicia un proceso de diferenciación en algunos aspectos de la implementación en los dos navegadores, proceso que culminaría con el nacimiento de JScript, nombre con el que Microsoft denomina a su versión de JavaScript. En la actualidad Microsoft ha desarrollado su JScript.net.

Las principales características de este lenguaje son:

- Es un lenguaje interpretado.
- No necesita compilación.
- Multiplataforma.
- Lenguaje de alto nivel.
- Admite programación estructurada.
- Basado en objetos.
- Maneja la mayoría de los eventos que se pueden producir sobre la página web.
- No se necesita ningún kit o entorno de desarrollo.

A diferencia de Java, JavaScript no dispone de elementos para crear interfaces de usuario propias para los programas y tiene que utilizar para ello los formularios de HTML a través de los denominados manejadores de eventos.

## **Java**

Java es un lenguaje de programación clásico en cuanto a que requieren un proceso de compilación. El código compilado puede ser integrado en la página web para que sea ejecutado por el cliente.

El nacimiento formal del lenguaje se sitúa en enero de 1996 con el lanzamiento por parte de la empresa creadora, Sun Microsystems, del JDK 1.0 (Java Development Kit). Este entorno de desarrollo Java puede obtenerse de forma totalmente gratuita a través de Internet (<http://www.javasoft.com>) e incorpora los elementos básicos necesarios para el desarrollo de aplicaciones Java.

Con Java se pueden crear dos tipos de programas:

- *Applets*: programas que se integran en las páginas web y que, residiendo en el servidor, son ejecutados por el cliente. La ejecución necesita de la interpretación del código compilado por el software cliente.
- *Aplicaciones*: programas autónomos que se pueden ejecutar en cualquier equipo. En este último caso puede optarse por generar código compilado similar al de los *applets* y que para su ejecución necesita de un intérprete o código compilado ejecutable directamente como en cualquier otro lenguaje de programación.

En el caso de los applets, el código fuente no se incrusta directamente en el documento HTML, sino que lo que se añade es un código binario resultado de la compilación, el denominado JBC (Java Byte Code). Esto permite proteger el código fuente, aunque hasta cierto punto, ya que las particularidades de este código compilado hacen que sea factible el proceso inverso, es decir, la decompilación, recuperar el código fuente a partir del compilado. En la propia Internet pueden encontrarse programas capaces de hacerlo. La razón de todo esto está en el hecho de que para conseguir la portabilidad de los programas el código compilado es un código que se encuentra a mitad de camino entre un código fuente y un código objeto fuertemente dependiente de una plataforma. Es por ello que se suele decir de Java que es un lenguaje que combina la flexibilidad de los lenguajes interpretados y el poder de los compilados.



Por supuesto, Java es un lenguaje con unas altas prestaciones, mucho mayores que las de lenguajes interpretados. Algunas de sus características son:

- Es un lenguaje orientado a objeto.
- Admite programación concurrente.
- Dispone de clases de objetos para la generación de interfaces gráficas de usuario.
- Tiene prestaciones multimedia.
- Resulta un lenguaje familiar, al tener una sintaxis similar al C++, aunque eliminando algunos de los problemas más engorrosos del lenguaje C: el uso de punteros, la gestión de la memoria y el control de accesos a los elementos de arrays.
- Es un lenguaje simple, robusto y seguro.
- A través de Internet se puede acceder a todo lo necesario para desarrollar applets Java.

## **VBScript**

VBScript es, al igual que JavaScript, un lenguaje basado en guiones que permite integrar programas directamente en el código HTML. Admite un doble uso, por un lado como lenguaje del lado del cliente, pero también como lenguaje del lado del servidor para la generación de páginas ASP.

Es un lenguaje desarrollado por Microsoft tomando como referente de sintaxis el VBA, Visual Basic para Aplicaciones. Por supuesto, no ofrece todas las funcionalidades de un entorno de desarrollo visual como Visual Basic, pero si se presenta como un herramienta poderosa y de fácil uso para generar páginas web interactivas.

### **1.4.3. TECNOLOGÍAS DE PROGRAMACIÓN DEL LADO DEL SERVIDOR**

Al igual que se hizo en la sección anterior, se presentan a continuación algunas de las tecnologías de programación del lado del servidor más conocidas. Dentro de esta categoría es en la que se sitúa el lenguaje PHP, que se estudiará con detalle en los próximos capítulos.

## **Programación CGI**

CGI son las siglas de *Common Gateway Interface* (Interfaz de Pasarela Común) y lo que define es un estándar para establecer la comunicación entre un servidor web y un programa. Esta interfaz define una forma cómoda y simple de ejecutar programas que se encuentran en la máquina en la que se aloja el servidor, a través de la definición de una serie de reglas que deben cumplir tanto las aplicaciones como los servidores para hacer posible la ejecución de los programas.

Al tratarse de una interfaz, no existe ningún tipo de dependencia con el lenguaje de programación empleado. Para desarrollar programas CGI se puede utilizar cualquier lenguaje. Los más habituales son: C, C++, Fortran, Perl, Tcl, Visual Basic, AppleScript. Los lenguajes interpretados como Tcl y Perl, tienen mayor facilidad de mantenimiento y depuración, presentan ventajas de seguridad, pero resultan más lentos. Los lenguajes compilados (C, C++,...), por su parte, son mucho más rápidos. En el caso de CGI la velocidad de ejecución es importante, ya que habrá que sumar el tiempo de ejecución al tiempo de espera de red y a la velocidad de transmisión.

## **ASP: Páginas de Servidor Activas**

ASP (Active Server Pages) es la tecnología diseñada por Microsoft para facilitar la creación de sitios web con una mayor sencillez que la empleada en la programación CGI. El principal inconveniente es la fuerte dependencia del entorno Microsoft, ya que requiere un servidor web de Microsoft, como puede ser el Internet Information Server (IIS) o el Personal Web Server (PWS).

Para utilizar la tecnología ASP sobre otros servidores, por ejemplo servidores Unix, se necesita un software intérprete (Chilisoft, Instant ASP).

El núcleo de funcionamiento de ASP es una aplicación ISAPI (Internet Server API). Una aplicación ISAPI es una DLL de Windows que se ejecuta en el mismo espacio de direcciones que el servidor web y que puede soportar varias peticiones simultáneas.

ASP no es realmente un lenguaje como tal, el lenguaje usado en realidad para programar ASP es Visual Basic Script o Jscript (versión Microsoft de JavaScript).

## **Servlets y JSP: Páginas de Servidor Java**

Los *servlets* y Java Server Pages (JSPs) son dos métodos de creación de páginas web dinámicas en servidor usando el lenguaje Java. Se trata de tecnologías desarrolladas por la empresa Sun Microsystems.

Las JSP se diferencian de otras tecnologías del lado del servidor como los CGI o las ASP en dos aspectos principalmente: por un lado, los JSP y *servlets* se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de ordenador, siempre que tenga instalado esa máquina virtual. Por otro lado, un programa JSP se compila a un programa en Java la primera vez que se invoca, y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un *servlet*. De esta manera los *servlets* no se ejecutan cada vez que se recibe una petición, sino que persisten de una petición a la siguiente, lo que permite realizar operaciones como la conexión a bases de datos o manejo de sesiones de una manera más eficiente.

Un JSP es una página web con etiquetas especiales y código Java incrustado, mientras que un *servlet* es un programa que recibe peticiones y genera a partir de ellas una página web. En ambos casos se necesita un programa servidor que se encargue de recibir las peticiones, distribuir las entre los *servlets* y realizar las tareas de gestión propias de un servidor web. Estos programas suelen llamarse contenedores de *servlets* o *servlet engines*, y, entre otros, podrían citarse como ejemplos Resin, BEA Weblogic, JRun de Macromedia, Lutris Hendirá, o, quizás el más popular y conocido: Tomcat.

## **ColdFusion**

ColdFusion es una tecnología desarrollada inicialmente por Allarie, que en la actualidad pertenece a Macromedia. Es una herramienta sencilla de aprender y bastante potente que funciona sobre la mayoría de servidores web. Los scripts se desarrollan por medio de etiquetas al estilo HTML (ColdFusion en realidad se denomina Cold Fusion Markup Language -CFML-). Estas etiquetas se sitúan dentro del documento HTML y son ejecutadas por el servidor, de forma que el cliente solo ve el resultado, no el código.

Básicamente ColdFusion está formado por tres componentes: una aplicación servidor, un lenguaje de marcación (ColdFusion Markup Language, CFML) y un programa administrador. La aplicación servidor es la encargada de leer e interpretar las instrucciones que le son pasadas a través de páginas ColdFusion. Estas páginas se identifican por tener la extensión `.cfm` o `.cfc` y contienen etiquetas HTML y etiquetas específicas del lenguaje propio de ColdFusion: CFML.

Las etiquetas del lenguaje CFML tienen una sintaxis similar a las de HTML y se distinguen del resto por tener nombres que siempre comienzan con `cf`. Además de las etiquetas predefinidas, el programador puede crear nuevas etiquetas e incluso puede integrar código en otros lenguajes como C, C++ o Java.

Las aplicaciones ColdFusion pueden interactuar con cualquier base de datos que soporte ODBC o JDBC.

## **PHP**

PHP es un lenguaje interpretado del lado del servidor que surge dentro de la corriente denominada código abierto (*open source*). Se caracteriza por su potencia, versatilidad, robustez y modularidad. Al igual que ocurre con tecnologías similares, los programas son integrados directamente dentro del código HTML. En este libro se explicará en detalle la sintaxis y el funcionamiento de este lenguaje, de momento se realiza a continuación una breve comparativa con las otras tecnologías del lado del servidor descritas previamente.

Comparado con ASP, la principal ventaja de PHP es su carácter multiplataforma. Por otro lado, los programas en ASP resultan más lentos y pesados, y también menos estables. En los entornos Microsoft la ventaja de ASP es que los servidores web de Microsoft soportan directamente ASP sin necesidad de ninguna instalación adicional

Señalar también la existencia de herramientas que permiten convertir programas desarrollados en ASP al lenguaje PHP, una de las más conocidas es *asp2php*.

Comparando el lenguaje PHP con el lenguaje Perl, utilizado habitualmente en la programación CGI, puede decirse que PHP fue diseñado para desarrollo de scripts orientados a web, mientras que Perl fue diseñado para hacer muchas más cosas y debido a esto, se hace muy complicado. La sintaxis de PHP es menos confusa y más estricta, pero sin perder la flexibilidad.

En comparación con ColdFusion, PHP es más rápido y eficiente para tareas complejas de programación, además PHP resulta más estable y usa una menor cantidad de recursos. Por el contrario, ColdFusion posee un mejor gestor de errores, un buen motor de búsquedas, abstracciones de bases de datos y un gran número de funcionalidades para el procesamiento de fechas. Finalmente, ColdFusion no está disponible para todas las plataformas.

En definitiva, PHP es uno de los lenguajes más utilizados actualmente en el desarrollo de aplicaciones web y viene experimentado un constante crecimiento en su nivel de utilización en Internet. Este libro trata de humildemente contribuir a continuar con el proceso de difusión de esta tecnología.