

PHP

Creación de páginas Web dinámicas

Hernán Beati

Buenos Aires • Bogotá • México DF • Santiago de Chile

 **Alfaomega**

Beati, Hernán

PHP : creación de páginas Web dinámicas. - 1a. ed. - Buenos Aires : Alfaomega Grupo Editor Argentino, 2011.

400 p. ; 23x17 cm.

ISBN 978-987-1609-21-5

1. Informática. 2. Diseño Web. I. Título
CDD 004.678

Queda prohibida la reproducción total o parcial de esta obra, su tratamiento informático y/o la transmisión por cualquier otra forma o medio sin autorización escrita de Alfaomega Grupo Editor Argentino S.A.

Edición: Damián Fernandez

Corrección: Juan Manuel Arana y Silvia Mellino

Revisión de armado: Laura Lago

Diagramación de interiores: Iris Biaggini

Diseño de tapa: Iris Biaggini

Internet: <http://www.alfaomega.com.mx>

Todos los derechos reservados © 2011, por Alfaomega Grupo Editor Argentino S.A.
Paraguay 1307, PB, oficina 11

Queda hecho el depósito que prevé la ley 11.723

NOTA IMPORTANTE: La información contenida en esta obra tiene un fin exclusivamente didáctico y, por lo tanto, no está previsto su aprovechamiento a nivel profesional o industrial. Las indicaciones técnicas y programas incluidos han sido elaborados con gran cuidado por el autor y reproducidos bajo estrictas normas de control. Alfaomega Grupo Editor Argentino S.A. no será jurídicamente responsable por errores u omisiones, daños y perjuicios que se pudieran atribuir al uso de la información comprendida en este libro, ni por la utilización indebida que pudiera dársele.

Los nombres comerciales que aparecen en este libro son marcas registradas de sus propietarios y se mencionan únicamente con fines didácticos, por lo que Alfaomega Grupo Editor Argentino S.A. no asume ninguna responsabilidad por el uso que se dé a esta información, ya que no infringe ningún derecho de registro de marca. Los datos de los ejemplos y pantallas son ficticios, a no ser que se especifique lo contrario.

Empresas del grupo:

Argentina: Alfaomega Grupo Editor Argentino S.A.

Paraguay 1307 P.B. "11", Buenos Aires, Argentina, C.P. 1057

Tel.: (54-11) 4811-7183 / 8352

E-mail: ventas@alfaomegaeditor.com.ar

México: Alfaomega Grupo Editor S.A. de C.V.

Pitágoras 1139, Col. Del Valle, México, D.F., México, C.P. 03100

Tel.: (52-55) 5575-5022 – Fax: (52-55) 5575-2420 / 2490. Sin costo: 01-800-020-4396

E-mail: atencionalcliente@alfaomega.com.mx

Colombia: Alfaomega Colombiana S.A.

Carrera 15 No. 64 A 29, Bogotá, Colombia

PBX (57-1) 2100122 - Fax: (57-1) 6068648

E-mail: sciente@alfaomega.com.co

Chile: Alfaomega Grupo Editor S.A.

Dr. La Sierra 1437-Providencia, Santiago, Chile

Tel.: (56-2) 235-4248 – Fax: (56-2) 235-5786

E-mail: agechile@alfaomega.cl

A mis tres mujeres (mi esposa Katty y mis hijas Stephanie y Violeta), que soportaron pacientemente las horas que les robé para poder escribir este libro.

A Maximiliano Firtman, que siempre confió en mis posibilidades, haciéndome parte de sus equipos de docentes, y ahora brindándome la oportunidad de publicar este libro.

A Damián Fernández, de Alfaomega Grupo Editor, por su seguimiento y estímulo constante.

A todos mis alumnos, porque todo este esfuerzo sólo cobra sentido en la medida en que puedan aprovecharlo para su crecimiento profesional.

Hernán Beati



Mensaje del Editor

Los conocimientos son esenciales en el desempeño profesional. Sin ellos es imposible lograr las habilidades para competir laboralmente. La Universidad o las instituciones de formación para el trabajo ofrecen la oportunidad de adquirir conocimientos que serán aprovechados más adelante en beneficio propio y de la sociedad. El avance de la ciencia y de la técnica hace necesario actualizar continuamente esos conocimientos. Cuando se toma la decisión de embarcarse en una actividad profesional, se adquiere un compromiso de por vida: mantenerse al día en los conocimientos del área u oficio que se ha decidido desempeñar.

Alfaomega tiene por misión ofrecer conocimientos actualizados a estudiantes y profesionales dentro de lineamientos pedagógicos que faciliten su utilización y permitan desarrollar las competencias requeridas por una profesión determinada. Alfaomega espera ser su compañera profesional en este viaje de por vida por el mundo del conocimiento.

Esta obra contiene numerosos gráficos, cuadros y otros recursos para despertar el interés del estudiante y facilitarle la comprensión y apropiación del conocimiento.

Cada capítulo se desarrolla con argumentos presentados en forma sencilla y estructurada claramente hacia los objetivos y metas propuestas. Asimismo, cada uno de ellos concluye con diversas actividades pedagógicas para asegurar la asimilación del conocimiento y su extensión y actualización futuras.

Los libros de Alfaomega están diseñados para ser utilizados dentro de los procesos de enseñanza-aprendizaje y pueden ser usados como textos guía en diversos cursos o como apoyo para reforzar el desarrollo profesional.

Alfaomega espera contribuir así a la formación y el desarrollo de profesionales exitosos para beneficio de la sociedad.

Acerca del autor

No nací programador. Por esta razón, estoy convencido de que todas las personas pueden aprender a programar y me especializo en su capacitación.

Me formé pedagógicamente haciendo un magisterio de música hace veinte años; luego me dediqué a la literatura, más tarde al diseño gráfico y, hacia fines del siglo XX, me convertí en diseñador Web.

Conocer el lenguaje PHP fue lo que me hizo estudiar Análisis de Sistemas, y pronto pasé del diseño a la programación Web. Comencé a enseñar lo que sabía: fundé el campus virtual de SaberWeb, y fui ayudante de cátedra en la Universidad Maimónides, en la materia Programación Multimedial.

Tiempo después fui docente en los institutos ImageCampus y ITMaster. En la Universidad Tecnológica Nacional dicté varios años el curso Professional Webmaster y fui el creador del curso de Programador Web Avanzado, que aún se sigue dictando.

Sigo capacitándome continuamente: los últimos cursos que hice fueron sobre UML y patrones de diseño de objetos, y realicé talleres de Usabilidad y de Accesibilidad Web.

Siempre me gustó el modelo de desarrollo colaborativo, propio del software libre.

Hernán Beati, noviembre de 2010.

Contenido

CAPITULO 1

Más allá de HTML y CSS	1
No más páginas Web: aplicaciones Web!	1
Ganando nuevos mercados a dos competidores: diseñadores gráficos y empresas de sistemas	1
La lógica de PHP: un amigo invisible	2
Proceso de archivos HTML	2
Pre-proceso de archivos PHP	5
Páginas estáticas	7
Páginas dinámicas	7
Las bases de datos	7
MySQL: la base de los proyectos Web exitosos	8

CAPITULO 2

El ambiente PHP	13
Esas extrañas siglas: LAMP, MAMP, WAMP, xAMP	13
El hosting	14
Eligiendo un buen hosting con PHP y MySQL.....	14
El servidor local para pruebas.....	14
Cómo descargar un servidor de pruebas	15
Cómo instalar el servidor de pruebas ...	15
Cómo encender y apagar el servidor de pruebas	18

Cómo configurar el servidor de pruebas	21
Cómo crear y dónde colocar los archivos	23
Cómo navegar por archivos PHP usando el servidor de prueba	26

CAPITULO 3

Mezclando PHP y HTML	29
El concepto clave: completando las páginas HTML en el acto	29
Formas de que PHP escriba dentro de HTML	31
Apertura y cierre de la etiqueta PHP	32
Escribir en el código con la función print()	33
Cómo funciona el comando «echo»	35
Grandes bloques: heredoc.....	35
Copiando y pegando archivos enteros.....	37
Los rompecabezas de include y require	37
Diferencias entre include, require, include_once y require_once	40
Los comentarios	41

CAPITULO 4

Los almacenes de datos	43
Contenedores temporales y permanentes, de pocos y de muchos datos	43

Las variables: pocos datos, provisorios.....	45	Break (salir del switch).....	90
Las ventajas de declarar y usar variables.....	45	Operadores de comparación y lógicos.....	90
Elegiendo una convención para darles nombre.....	48	Operadores de comparación.....	90
El problema de las comillas.....	50	Operadores lógicos:.....	92
Concatenación.....	51	Tipos de validaciones.....	98
Las constantes: pocos datos que no cambiaremos.....	55	¿Está presente cada dato requerido?...	98
Una variable que no vamos a modificar.....	55	Si no está vacío.....	101
Includes de constantes.....	55	Si su valor está dentro de un rango....	102
Las matrices: muchos datos provisorios....	58		
Un paquete de variables.....	58	CAPITULO 7	
Índices numéricos.....	61	Identificación con cookies y sesiones	105
CAPITULO 5		Cookies: datos que identifican a un navegador.....	105
Enviando datos hacia el servidor	69	Almacenar variables en cookies.....	107
Herramientas para enviar datos: enlaces y formularios.....	69	Leer variables de cookies.....	109
Enlaces con variables en la URL.....	70	Borrar una variable o una cookie.....	113
Las matrices superglobales.....	72	Argumentos opcionales.....	114
Matrices que almacenan datos automáticamente.....	72	Ejemplo completo.....	117
Formularios.....	73	Ideas para aplicar cookies.....	119
Ventajas y limitaciones de enviar variables por el método "get".....	75	Sesiones: datos que identifican a un usuario.....	120
Ventajas y limitaciones de enviar variables por el método "post".....	76	¿Es mejor usar cookies o identificadores de sesión?.....	123
		Una clave que se compara contra un dato guardado en el servidor.....	124
		Cómo iniciar una sesión.....	125
		Cómo declarar variables de sesión.....	126
		Cómo transmitir el identificador de sesión sin cookies.....	127
		Cómo borrar una variable de sesión, o todas.....	130
		Cómo eliminar el archivo de sesión y la cookie.....	130
		La duración de las sesiones y la seguridad.....	132
		Ejemplo de acceso restringido.....	132
		Internacionalización usando sesiones.....	135
CAPITULO 6		CAPITULO 8	
Validaciones	77	Los bucles y los archivos de texto	139
Validando datos de formularios y enlaces.....	77	Tipos de bucles: for, while, do while, foreach.....	140
Los condicionales.....	78	Cómo repetir una acción en una cantidad fija de veces: el bucle "for".....	141
If (si...).....	78		
¿Qué hacer si responde que no es verdad?			
El «else» y el «elseif».....	80		
Else (si no).....	80		
Elseif (si no, si...).....	84		
Varias condiciones diferentes: el elseif.....	84		
Elegir entre valores conocidos de una única variable: el «switch».....	86		
Case (cada posible valor).....	87		
Default (valor por omisión).....	88		

El complemento perfecto de las matrices ordenadas y de los archivos de texto ..	144
El concepto de contador	145
El concepto de acumulador	147
Cómo recorrer una matriz con «foreach» ..	148
Cómo repetir algo una cantidad desconocida de veces: el «while»	150
Las funciones «list» y «each»	151
El bucle do-while.....	152
Condicionales dentro de bucles.....	153
La técnica de la señal o «flag»	154
Los archivos de texto	156
Un contenedor simple y permanente, para datos no muy confidenciales	156
1. Modos de apertura de archivos	157
2. Operaciones posibles: lectura, escritura, agregados de datos	158
3. Cierre de un archivo	159
Formas de leer datos desde un archivo de texto	159
Leer línea por línea con la función «file» ...	159
Función fpassthru.....	162
Función fread.....	163
Función fgetc	164
Función feof	165
Función fgets	167
Función fgetss	167
Cómo escribir y acumular datos en un archivo de texto	169
Funciones fputs y fwrite	170
Acumular datos sin borrar lo anterior	171

CAPITULO 9

Creando y usando funciones	173
Planificando nuestros sistemas Web.....	174
El modelo basado en la ejecución de tareas (funciones)	174
La función: una caja cerrada que procesa datos	176
Declarar una función	179
Nuestra primera función.....	180
Parámetros de entrada	182
Funciones con varios parámetros	184
Funciones sin parámetros.....	185

Devolviendo datos a la salida con return .	186
Las ventajas de evitar echo y print desde dentro de las funciones	189
Alcance de las variables en las funciones: locales y globales	191

CAPITULO 10

Funciones incorporadas más usadas ...	195
Funciones de manejo de caracteres	196
Limpiar espacios en blanco	196
Función trim	196
Función ltrim	197
Función rtrim o chop	197
Comparar evitando errores: strtolower y strtoupper	197
Contar la cantidad de letras de un texto ..	198
Obtener «partes» de una cadena de caracteres.....	199
Función substr	199
Función strpos.....	201
Función strstr	203
Buscar algo en un texto con preg_match	204
Buscar y reemplazar: str_replace	207
Dividir o ensamblar un texto con delimitadores: explode e implode	207
Convertir saltos de línea a breaks HTML..	209
Funciones printf y sprintf	211
Funciones de fecha y hora	213
El concepto de timestamp	213
Obtener fecha y hora actual con la función time.....	213
Convirtiendo de timestamp a formatos más humanos.....	214
Otra forma de mostrar fechas y horas: la función date.....	215
Zonas horarias.....	217
De día, mes y año, a valor de timestamp: la función mktime.....	218
Validar si una fecha es correcta con checkdate	219
Funciones de envío de correos electrónicos..	220
Un servidor de correos en el hosting	220
Envío básico.....	220
Casos prácticos	221

Formulario de consultas.....	222	BLOB	262
Recomendar una página.....	223	TINYBLOB, MEDIUMBLOB y	
Agregar un remitente (From).....	225	LONGBLOB	262
Enviando mensajes HTML con remitente		ENUM.....	262
y otras cabeceras adicionales.....	226	SET.....	263
Saltos de línea en texto plano:.....	226	Datos de fecha y hora	264
Cuerpo del mensaje con código HTML:.....	227	DATE.....	264
		DATETIME	264
		TIME.....	264
		TIMESTAMP	264
		YEAR	265
CAPITULO 11		Atributos de los campos	266
Creando bases de datos	231	¿Null o Not Null?.....	266
El almacén de datos más potente para		Valor predeterminado (default)	266
nuestros sitios Web.....	231	Ordenamiento binario	267
Diferencia entre archivos de texto y bases		Índices	267
de datos: el lenguaje SQL.....	232	PRIMARY Key y Auto_increment	268
Programas que utilizaremos:.....	233	UNIQUE	269
Conceptos fundamentales: base, tabla,		FULLTEXT	269
registro y campo.....	234		
Base de datos.....	234	CAPITULO 12	
Tablas.....	234	Llevando datos de la base a las	
Campos.....	235	 páginas	271
Registros	235	Cómo leer datos desde una base con PHP.....	271
Creando bases y tablas con phpMyAdmin	236	Probar la conexión.....	272
Seguridad.....	238	La orden "SELECT": entendiendo	
Crear una base de datos	242	el lenguaje SQL	273
Crear una tabla.....	245	Ejecutar la orden y obtener	
Proceso de altas, bajas y modificaciones.....	248	una respuesta.....	275
Dar de alta un registro	250	Integrar los datos al código HTML.....	276
Ver un listado con datos	252	Complementos de la orden SELECT del	
Modificar un registro	253	lenguaje SQL	279
Borrar un registro	253	El condicional WHERE.....	280
Copia de seguridad: cómo exportar		Operadores para utilizar con WHERE..	281
una tabla y volver a importarla.....	254	Ordenando los resultados.....	283
Los tipos de datos más usados	256	Limitando la cantidad de resultados	283
Datos numéricos	257	Seleccionando valores no repetidos.....	284
Numéricos enteros	257	Funciones estadísticas	284
Valores sin signo.....	258	Funciones propias para mostrar datos.....	285
Numéricos con decimales.....	259	Menú de selección dinámico.....	291
Datos alfanuméricos.....	259	Generando tablas, filas y datos	
CHAR.....	260	de tabla	293
VARCHAR	260	Los botones de radio y las casillas de	
BINARY y VARBINARY.....	261	verificación.....	294
TEXT	261		
TINYTEXT, MEDIUMTEXT y LONGTEXT	261		

CAPITULO 13**Llevando datos de las paginas****a la base** 297

Cómo escribir datos en una base desde PHP 297

Creando el formulario 299

Conexión a MySQL..... 300

La orden "INSERT" del lenguaje SQL 301

Ejecutar la consulta..... 302

Filtrar los datos de formularios: evitando inyección SQL..... 303

Cómo eliminar datos de una base con PHP... 307

La orden "DELETE" del lenguaje SQL.. 311

Cómo modificar datos de una base con PHP. 314

El formulario más complicado: la actualización..... 314

La orden "UPDATE" del lenguaje SQL 326

Radiografía de un sistema con back-end y front-end 328

Front-end: el punto de vista del usuario ... 329

Listado de tareas del administrador..... 331

APÉNDICE 1**Programacion orientada a objetos** 333

Otro paradigma, otras técnicas..... 333

clases, objetos, instancias, métodos y propiedades..... 334

Cómo definir una clase, propiedades y métodos 336

Cómo crear una instancia..... 338

Métodos getter y setter 344

Método getter..... 344

Método setter..... 344

Definiendo la visibilidad..... 345

Pública..... 345

Privada..... 345

Protegida 345

Cómo aplicar el concepto de herencia 346

Atributos de clase..... 347

Constantes 347

Static..... 348

Versiones anteriores 349

Recursos para utilizar libremente miles de clases pre-armadas 350

Creando un mapa de Google en un instante..... 351

Qué hace cada método 353

Los parámetros posibles de addControl..... 354

APÉNDICE 2**Adaptando software libre** 357

Licencias para no reinventar la rueda..... 359

Por qué PHP tiene tantos proyectos listos para usar..... 359

Sistemas para todos los rubros y clientes..... 360

Principales CMS: portales, ecommerce, gestión de proyectos 360

Criterios para elegir entre proyectos similares..... 362

Instalando un sistema para blogs..... 363

Modificaciones básicas..... 367

Estructura para modificar un plugin 372

Cómo crear un plugin..... 373

Prólogo

En 1994 un programador nacido en Groenlandia, llamado Rasmus Lerdorf (<http://lerdorf.com>), desarrolló un código que le ayudaría a crear su página Web personal de manera más sencilla. Lo llamó Personal Home Page Tools (PHP Tools) o herramientas para páginas iniciales personales. De las primeras tres palabras en inglés surge el nombre del lenguaje que finalmente se liberó al público, gratis, en 1995.

Quince años después, el mundo de la Web ha cambiado drásticamente. La evolución y difusión de PHP en el mundo del desarrollo Web ha ido mucho más allá de lo que Rasmus pudo imaginar; se trata de un mundo del que ahora podrás ser parte.

Dos años más tarde, junto a otras personas, se reescribe parte del código del lenguaje y se lanza la versión de PHP que ha llevado el lenguaje al estrellato: PHP 3. Un lenguaje simple, rápido y dinámico que permite crear páginas Web interactivas con muy poco código.

En ese momento deciden que el nombre Personal Home Page ya le quedaba un poco corto al lenguaje y deciden cambiar el significado de las siglas. Así es que hoy PHP significa "PHP Hypertext Preprocessor". No es un error de imprenta: la "pe" de PHP significa PHP. Es una sigla recursiva (un truco de programadores) y el resto del nombre significa "pre-procesador de hipertexto". Es un pre-procesador porque se ejecuta antes que el navegador y trabaja principalmente sobre hipertexto, que es el concepto subyacente de los documentos HTML.

Con los años igualmente se ha ganado su propio nombre. PHP es, simplemente, PHP. Es tan importante en la Web como lo es HTML. Es un lenguaje fácil de aprender, simple de usar, potente, rápido, gratuito, de código abierto y utilizado en más de la mitad de todos los sitios Web del mundo.

Sólo para ejemplificar la potencia del lenguaje, mencionaremos que el sitio más importante y con más visitas hoy en el mundo, Facebook, está desarrollado con PHP.

Tuve la oportunidad de conocer personalmente a Rasmus en Madrid en una conferencia sobre optimización extrema de PHP y me ha quedado muy presente una anécdota que me gustaría compartir contigo. Rasmus tuvo la oportunidad de analizar el código fuente PHP de las primeras versiones de Facebook. El código era un desastre, pésimamente programado, con errores por todos lados.

Y a pesar del desastre que él detectó que era esa Web desde el lado del código, ¡funcionaba bien! Y miren en lo que Facebook se ha convertido: una empresa billonaria. La moraleja de la anécdota es que la gran ventaja de PHP es su ca-

pacidad de funcionar sin problemas en cualquier circunstancia y de poder tener una Web lista en muy poco tiempo.

Rasmus, un poco exagerando para hacer entender el objetivo de su moraleja, comentó que los proyectos más exitosos en la Web no son los mejor programados, los que siguen las mejores prácticas o los que son desarrollados por académicos. Son aquellos que implementan las mejores ideas y lo hacen lo más rápido posible; por eso PHP es el lenguaje ideal para la Web.

Luego de esta introducción, estarás seguro con muchas ganas de empezar a trabajar con PHP. Y qué mejor que hacerlo de la mano de Hernán Beati, un excelente profesional y profesor que conozco hace ya diez años y recomiendo plenamente para tu viaje en el mundo de este apasionante lenguaje.

¡A programar!

Lic. Maximiliano Firtman
Director | ITMaster Professional Training
Adobe Community Champion
Autor de "AJAX, Web 2.0 para profesionales".
www.twitter.com/firt

Antes de comenzar a leer:

Para comodidad del lector, el código utilizado en el libro se encuentra disponible para ser descargado desde:

<http://www.alfaomega.com.mx/archivosadicionales>



Alfaomega e ITMaster Professional Training te dan la posibilidad de que certifiques tus conocimientos y experiencias adquiridos como lector de este libro. Su aprobación te permitirá tener la certificación en Programación PHP Inicial.

Luego de la obtención del certificado, podrás continuar tu formación en la carrera corta de Programador Web, en los másters de especialización Programador Experto PHP y Experto en Mobile Web y en los cursos cortos presenciales y online disponibles en todo el mundo.

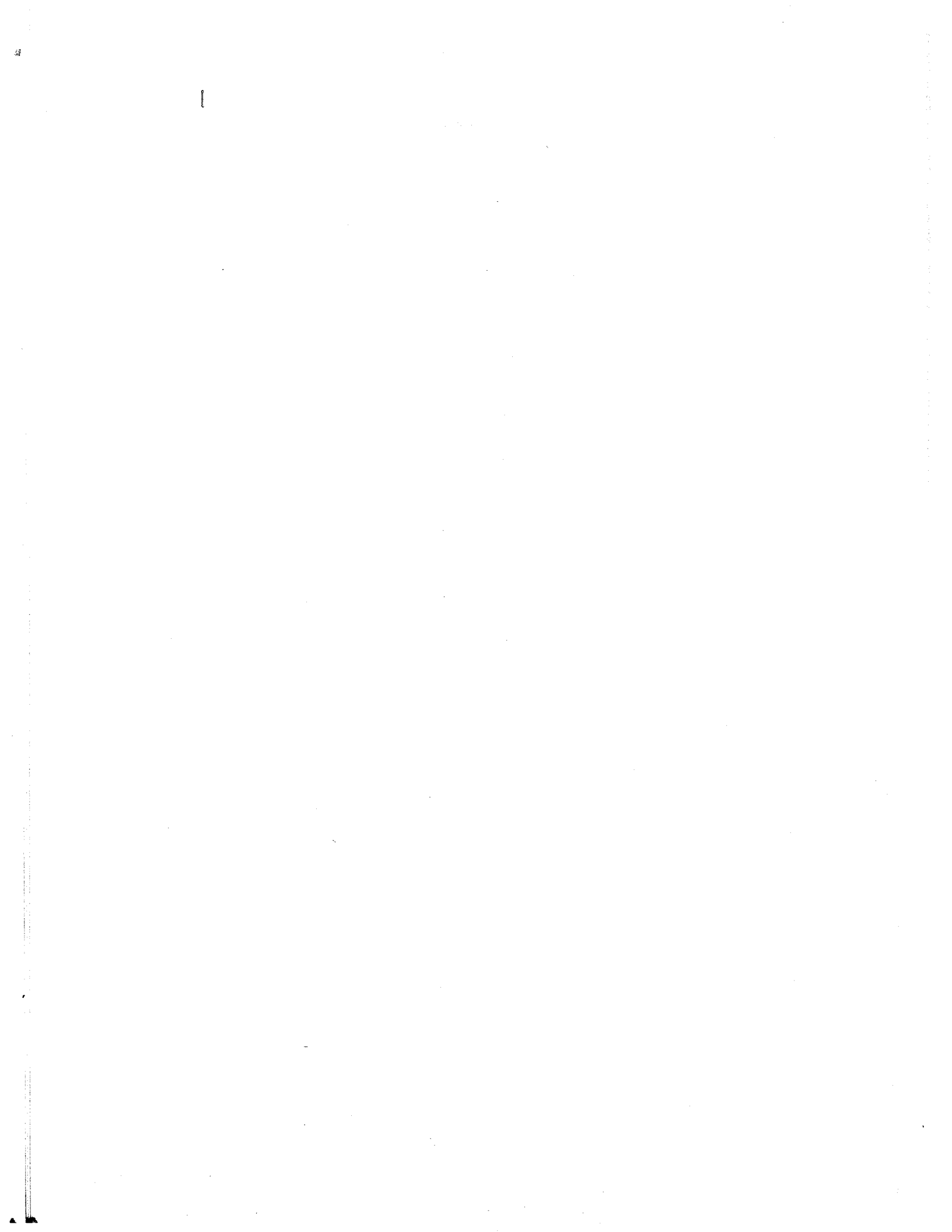
Para dar la evaluación de certificación o recibir mayor información sobre este servicio, ingresa en el sitio Web o envíanos un e-mail a la dirección correspondiente a tu país

Argentina: www.itmaster.com.ar - info@itmaster.com.ar

México: www.itmaster.com.mx - info@itmaster.com.mx

España: www.itmaster.es - info@itmaster.es

Otro país: www.itmaster.la - info@itmaster.la



1

MÁS ALLÁ DE HTML Y CSS

¡No más páginas Web: aplicaciones Web!

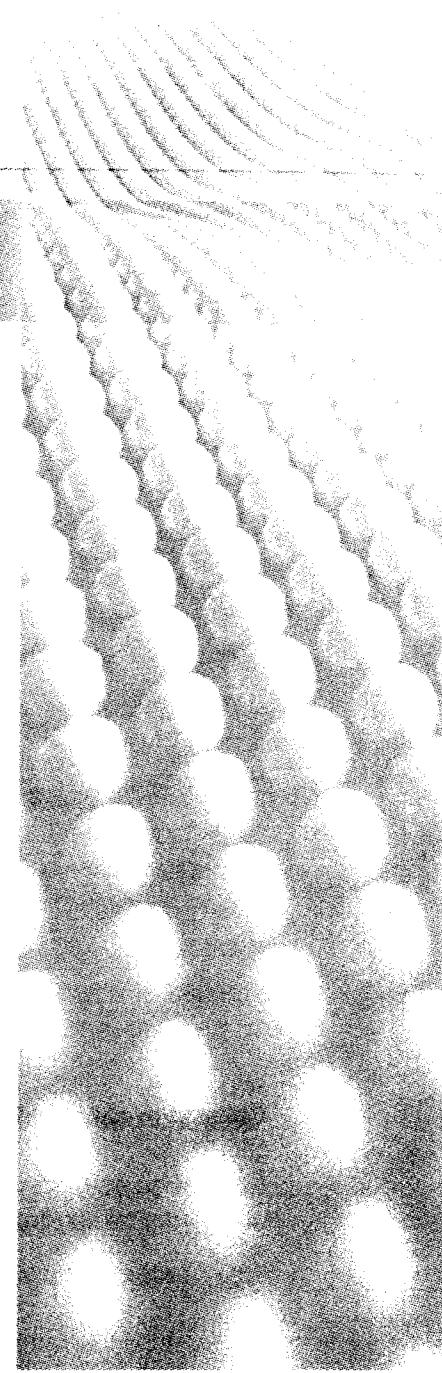
"No es la Programación, son los Negocios". Eso es PHP.

Dominar el lenguaje PHP amplía nuestros horizontes profesionales como diseñadores o programadores, y nos convierte en creadores de **Aplicaciones Web**. Nos lleva de la mano a un mundo de comercio electrónico, redes sociales, intranets, portales de noticias y entretenimientos, un mundo "mágico" en el que podemos acceder gratuitamente a miles de sistemas completos prearmados, listos para usar (y para vender a nuestros nuevos clientes).

Nos abre un nuevo mercado, donde los clientes ya no están tan interesados en el diseño (aunque puede aportar su encanto), sino en las **funcionalidades**, para que, a través de un navegador, las personas puedan hacer alguna tarea concreta en su sitio Web.

Ganando nuevos mercados a dos competidores: diseñadores gráficos y empresas de sistemas

PHP nos despega de nuestros antiguos competidores (diseñadores gráficos o programadores de aplicaciones de escritorio) y nos lleva a un nuevo mercado, en el que ofrecemos soluciones Web a comercios, empresas de diversos tamaños, profesionales, instituciones educativas, medios de difusión.



Ahora competimos con empresas de sistemas, con la ventaja de la rapidez y economía de nuestras soluciones PHP.

LENGUAJES:	HTML/CSS	PHP/MySQL	Otros (Java, .Net)
COMPETIDORES:	Diseñadores Gráficos	Programadores Web	Empresas de Sistemas
TAREA PRINCIPAL:	Decorar páginas (no saben programar)	Adaptar sistemas prearmados rápida y económicamente	Hacer sistemas a medida desde cero (caros y de largo plazo)
NIVEL DE PRESUPUESTOS:	Cientos	Miles / Decenas de miles	Decenas a centenas de miles

Cuadro 1.1. Mercado de los sistemas prearmados

Eso explica por qué se propaga con tanta velocidad el conocimiento de PHP entre diseñadores y programadores de otros lenguajes. Es la clave para llevar a cabo cualquier proyecto que trascienda las páginas Web HTML estáticas.

Y el detalle fundamental: con PHP, nuestros presupuestos pueden llegar a tener uno o incluso dos ceros más que los presupuestos que hacíamos como diseñadores o programadores de otros lenguajes.

Además, PHP es fácil de aprender.

La lógica de PHP: un amigo invisible

¿Qué es PHP? PHP no se ve. Es “transparente”, invisible. Por esta razón, es difícil explicar qué es y cuál su funcionamiento. Sin embargo, lo intentaremos.

Éste es el concepto más abstracto del libro, pero es imprescindible para entender qué hace PHP.

PHP es una sigla, un acrónimo de “**PHP: Hypertext Preprocessor**”, o sea, “Pre-procesador de Hipertexto marca PHP”. El hecho de que sea un “pre” procesador es lo que marca la diferencia entre el proceso que sufren las páginas Web programadas en PHP del de aquellas páginas Web comunes, escritas sólo en lenguaje HTML.

Para llegar a entender qué es un pre-procesador, examinaremos primero cuál es la diferencia entre el proceso de una página Web normal (HTML) y el “pre” proceso de una página escrita en lenguaje PHP.

Proceso de archivos HTML

¿Cuál es el camino que sigue una página Web común (escrita en lenguaje HTML) desde que escribimos su dirección en nuestro navegador hasta que la vemos en nuestra pantalla?

1. Comenzamos escribiendo en el navegador la URL deseada y pulsamos **enter** (o pulsamos un enlace con el *mouse*); en ambos casos, la barra de direcciones nos muestra la URL del archivo HTML que nuestro navegador está solicitando:

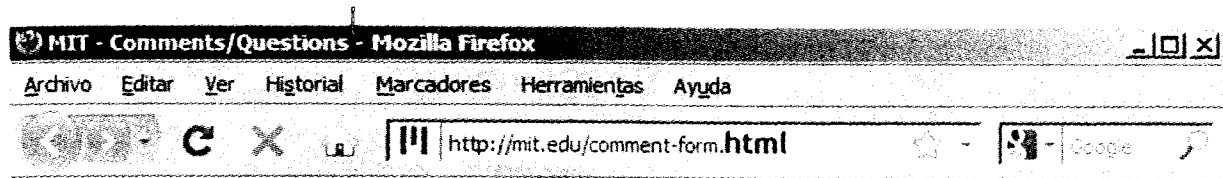


Figura 1-1. URL de un archivo HTML.

2. En ese momento, el navegador envía una petición que solicita esa página. Ese pedido "viaja" desde nuestro navegador hasta la máquina *hosting* que hospeda el archivo requerido. Pero el navegador no remite únicamente el pedido del archivo que necesita, sino que lo acompaña con un número que nos identifica inequívocamente: nuestra dirección IP.

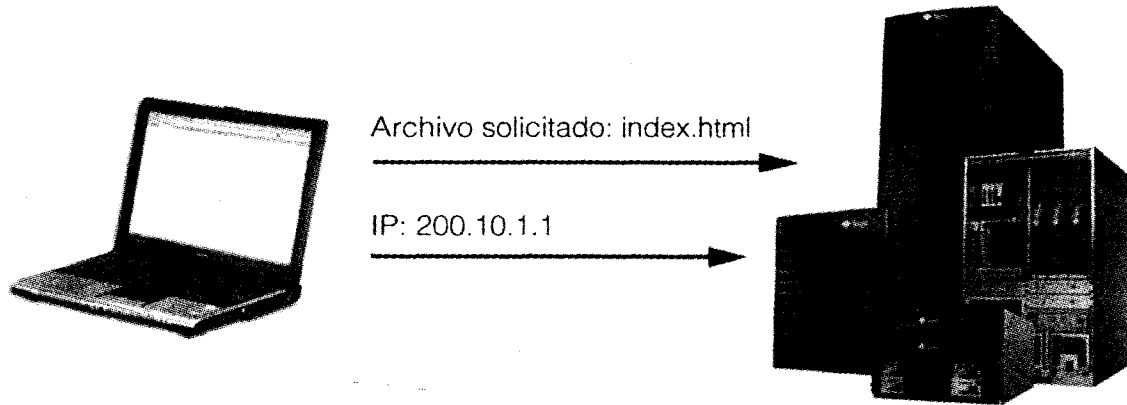


Figura 1-2. El navegador solicita un archivo y envía nuestra dirección IP.

Podemos compararlo con un *delivery* de pizza a domicilio; para recibir el pedido, le decimos al telefonista dos datos: "cuál gusto de pizza" queremos y "a qué dirección" nos la debe enviar.

3. Cuando el pedido llega al *hosting* indicado, un programa denominado servidor Web que está encendido en esa máquina, recibe el pedido y va a buscar el archivo solicitado en el disco rígido.

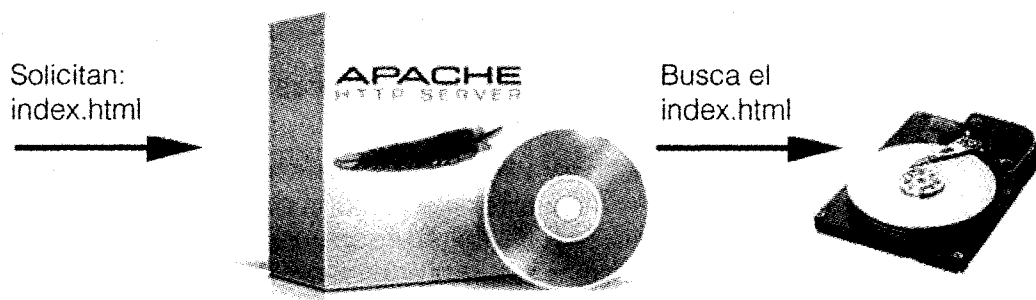


Figura 1-3. El servidor Web busca en el disco rígido del *hosting* el archivo solicitado.

El rol del servidor Web es similar al del empleado de la pizzería que atiende al teléfono y va a buscar el pedido a la cocina de la pizzería.

4. Ese servidor Web, una vez que localizó el archivo solicitado, envía, entrega o "sirve" (de ahí su nombre: "servidor") el archivo al navegador que se había quedado esperando una respuesta en la dirección IP que lo identifica.

Envía el
index.html

a la IP que
hizo el pedido

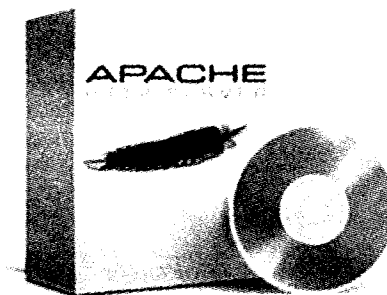
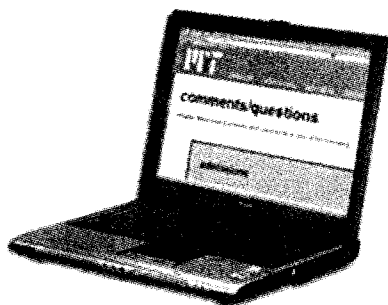


Figura 1-4. El servidor Web envía el archivo solicitado a la dirección IP del navegador.

Equivaldría al viaje de la moto que nos trae la pizza hacia nuestra casa.

- Una vez que llegó el archivo hasta nuestro navegador, éste se encarga de interpretar los contenidos de ese archivo de texto y código HTML, armando cada elemento (textos, tablas, colores) de la página recibida en nuestra pantalla para que la podamos leer.

Recibe el código HTML del index.html



Y lo interpreta,
mostrando
el resultado
en pantalla

```
<li
class="links"><a
href="http://web.mit.
edu/aboutmit/">about
MIT</a></li>
<li
class="links"><a
href="http://web.mit.
edu/admissions
```

Figura 1-5. El navegador interpreta el código HTML y muestra el resultado.

En la metáfora de la pizza, llega el momento de recibir la pizza y servirla en nuestra mesa, lista para comer.

Podemos mencionar algunas conclusiones que identifican el proceso de archivos HTML comunes:

- Nuestro **navegador** tiene la capacidad de pedir archivos a distintos servidores Web, y a su vez, de entender y “descifrar” esos archivos recibidos, cuyo contenido es código HTML. Nuestro navegador es un programa que todo el tiempo realiza pedidos de archivos (peticiones) y recibe un archivo HTML como respuesta, que luego muestra a los seres humanos. Pide archivos y los muestra, pide y muestra...
- El **servidor Web** es un programa instalado en los *hostings*, que todo el tiempo recibe pedidos de navegadores (se los llama peticiones), y le entrega a esos navegadores el archivo HTML solicitado.
- Este diálogo entre un navegador y un servidor Web sigue las reglas de un **protocolo** (una convención, un estándar) denominado HTTP (**H**yper**T**ext **T**ransfer **P**rotocol, o Protocolo de Transferencia de HiperTexto).

Todo esto sucede cada vez que queremos ver un archivo HTML común.

Pre-proceso de archivos PHP

Pero, ¿cuál es el camino que sigue una página Web cuya extensión es .php desde que escribimos su dirección en nuestro navegador hasta que la vemos?

Cuando la extensión del archivo solicitado es .php, se introduce un elemento diferente en este circuito:

1. Hacemos el pedido de ver una página con extensión .php desde nuestro navegador:

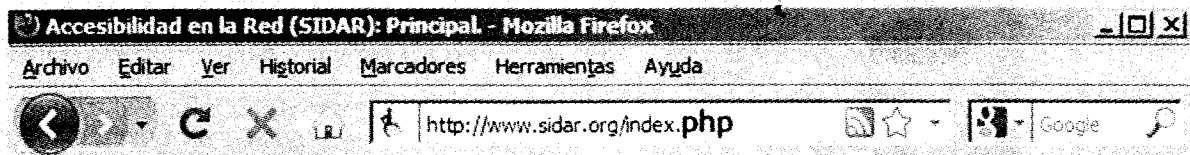


Figura 1-6. URL de un archivo PHP.

2. El programa servidor Web instalado en el *hosting* recibe nuestro pedido y, de inmediato, detecta que el archivo solicitado tiene extensión .php y, por lo tanto, deriva el pedido a otro programa que está encendido en esa misma máquina *hosting*, que se denomina **intérprete de PHP** (es una especie de "ser mágico", cuya presencia es muy difícil intuir, y que debemos acostumbrarnos a imaginar que "está ahí" para poder programar correctamente en PHP).

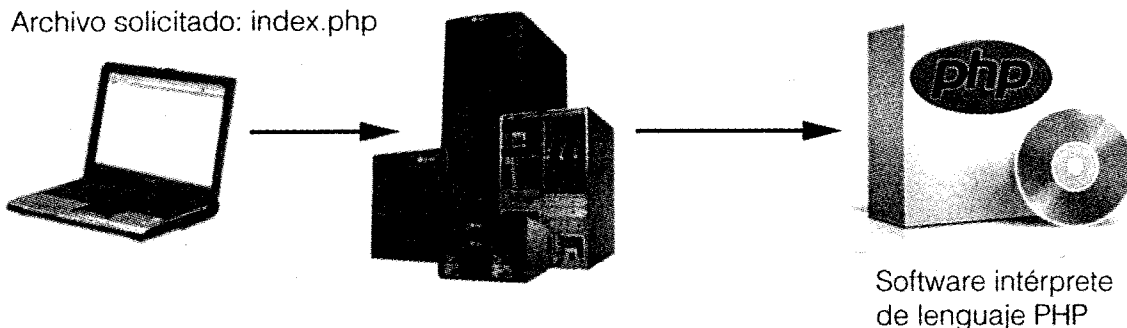


Figura 1-7. El servidor Web le pasa el pedido al intérprete de PHP.

3. Este programa intérprete de PHP busca en el disco rígido del *hosting* el archivo .php que fue solicitado, y comienza a leer su código línea por línea, buscando determinadas "marcas" o etiquetas que nosotros, como programadores, hemos dejado escritas y que contienen órdenes destinadas a ese programa intérprete de PHP.

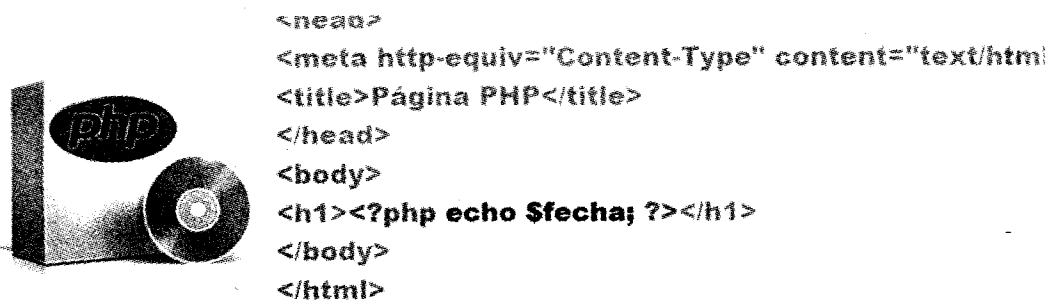


Figura 1-8. El intérprete de PHP busca las marcas con órdenes para él.