

# PHP



2a edición



# El gran libro de PHP

## Creación de páginas Web dinámicas

Cubre las versiones 4, 5 y 6

Hernán Beati



**PHP**  
**Creación de páginas Web dinámicas**  
**2ª ed.**

**Hernán Beati**



# **PHP**

## **Creación de páginas Web dinámicas**

### **2ª ed.**

**Hernán Beati**

 **Alfaomega**

 **marcombo**  
ediciones técnicas

*PHP. Creación de páginas web dinámicas*  
Hernan Beati

Derechos reservados © Alfaomega Grupo Editor Argentino, 2015. Ciudad Autónoma de Buenos Aires. Segunda edición: Alfaomega Grupo Editor Argentino, 2015

Segunda edición: MARCOMBO, S.A. 2016

© 2016 MARCOMBO, S.A.  
[www.marcombo.com](http://www.marcombo.com)

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

ISBN: 978-84-267-2273-7

D.L.: B-25204-2015

Impreso en Ulzama Digital SL  
*Printed in Spain*

## 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 vida 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 ofrecerles a estudiantes y profesionales conocimientos actualizados 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.

Alfaomega hace uso de los medios impresos tradicionales en combinación con las tecnologías de la información y las comunicaciones (TIC) para facilitar el aprendizaje.

Libros como éste tienen su complemento en una página Web, en donde el alumno y su profesor encontrarán materiales adicionales.

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.

Los libros de Alfaomega están diseñados para ser utilizados dentro de los procesos de enseñanza-aprendizaje, y pueden ser usados como textos para 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.

Al finalizar mi adolescencia, comencé mi formación pedagógica cursando un magisterio de música (que nunca ejercí); pero paralelamente me dediqué profesionalmente al diseño gráfico durante casi una década, y ya hacia fines del siglo XX, me convertí en diseñador Web.

Conocer el lenguaje PHP en el año 2000 fue lo que me hizo estudiar Análisis de Sistemas y la Licenciatura en Informática Educativa y pronto, me especialicé en programación para la Web, realizando numerosas aplicaciones Web, trabajando tanto para empresas como freelance.

Desarrollé mi perfil docente desde el año 2002, cuando fundé el campus virtual de SaberWeb y comencé a enseñar en los institutos ITMaster, Image Campus y Dotzero, en las carreras terciarias de Escuela Da Vinci e IMAGE, y para graduados de las carreras de Diseño en la FADU de la Universidad de Buenos Aires. En la Universidad Tecnológica Nacional (FRBA), dicté el curso Professional Webmaster y fui el creador del curso de Programador Web Avanzado, que aún se sigue dictando.

Desde 2013, soy profesor titular de las materias Programación Multimedial II y III en la Universidad Maimónides, dentro de la Licenciatura en Tecnología Multimedial, donde investigamos sobre la base de PHP y MySQL la creación de servicios Web para Apps móviles y videojuegos, y creamos aplicaciones Web orientadas a objetos.

Sigo capacitándome continuamente: los últimos cursos que hice en esta área fueron sobre PHP orientado a objetos, y backend de aplicaciones móviles y videojuegos.

Siempre me gustó el modelo de desarrollo colaborativo, propio del software libre, del que PHP es un perfecto exponente.

Hernán Beati, julio de 2015



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

**Hernán Beati**

## **Agradecimientos**

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 solo cobra sentido en la medida en que puedan aprovecharlo para su crecimiento profesional.

Hernán Beati

# Contenido

Prólogo.....	XIII	¿Qué hacer si responde que no es verdad? El else y el elseif.....	94
Plataforma de contenidos interactivos.....	XVI		
<b>CAPÍTULO 1</b>			
<b>MÁS ALLÁ DE HTML Y CSS.....</b>	<b>1</b>		
¡No más páginas Web: aplicaciones Web .....	1		
<b>CAPÍTULO 2</b>			
<b>EL AMBIENTE PHP.....</b>	<b>15</b>		
Esas extrañas siglas: LAMP, MAMP, WAMP, xAMP.....	15		
<b>CAPÍTULO 3</b>			
<b>MEZCLANDO PHP Y HTML.....</b>	<b>35</b>		
El concepto clave: completando las páginas HTML en el acto .....	35		
<b>CAPÍTULO 4</b>			
<b>LOS ALMACENES DE DATOS .....</b>	<b>51</b>		
Contenedores temporales y permanentes, de pocos y de muchos datos.....	51		
Las variables: pocos datos, provisorios .....	53		
Las constantes: pocos datos que no cambiaremos.....	65		
Las matrices: muchos datos provisorios....	69		
<b>CAPÍTULO 5</b>			
<b>ENVIANDO DATOS HACIA EL SERVIDOR.....</b>	<b>81</b>		
Herramientas para enviar datos: enlaces y formularios.....	81		
<b>CAPÍTULO 6</b>			
<b>VALIDACIONES.....</b>	<b>91</b>		
Validando datos de formularios y enlaces.	91		
Las condicionales.....	92		
<b>CAPÍTULO 7</b>			
<b>IDENTIFICACIÓN CON COOKIES Y SESIONES .....</b>	<b>123</b>		
Cookies: datos que identifican a un navegador .....	123		
Sesiones: datos que identifican a un usuario.....	141		
<b>CAPÍTULO 8</b>			
<b>LOS BUCLES Y LOS ARCHIVOS DE TEXTO .....</b>	<b>163</b>		
Recorriendo línea por línea la información almacenada .....	163		
Tipos de bucles: for, while, do while, foreach.....	164		
Condicionales dentro de bucles .....	179		
Los archivos de texto.....	182		
Formas de leer datos desde un archivo de texto.....	186		
Cómo escribir y acumular datos en un archivo de texto .....	198		
<b>CAPÍTULO 9</b>			
<b>CREANDO Y USANDO FUNCIONES .....</b>	<b>203</b>		
Reutilizando nuestros códigos.....	203		
Planificando nuestros sistemas Web.....	205		
La función: una caja cerrada que procesa datos.....	207		
Declarar una función.....	210		
<b>CAPÍTULO 10</b>			
<b>FUNCIONES INCORPORADAS MÁS USADAS .....</b>	<b>229</b>		

Manejo de caracteres, fechas y envío de correos .....	229
Funciones de manejo de caracteres .....	230
Funciones de fecha y hora .....	249
Funciones de envío de correos electrónicos.....	257

#### CAPÍTULO 11

<b>CREANDO BASES DE DATOS</b> .....	269
El almacén de datos más potente para nuestros sitios Web.....	269
Diferencia entre archivos de texto y bases de datos: el lenguaje SQL.....	270
Conceptos fundamentales: base, tabla, registro y campo .....	272
Creando bases y tablas con phpMyAdmin.....	275
Proceso de altas, bajas y modificaciones ..	289
Los tipos de datos más usados.....	297
Atributos de los campos .....	309

#### CAPÍTULO 12

##### LLEVANDO DATOS DE LA BASE A

<b>LAS PÁGINAS</b> .....	315
Cómo leer datos desde una base con PHP.....	315
PHP.....	315
Complementos de la orden SELECT del lenguaje SQL.....	324
Funciones propias para mostrar datos .....	331

#### CAPÍTULO 13

##### LLEVANDO DATOS DE LAS PÁGINAS

<b>A LA BASE</b> .....	345
Cómo escribir datos en una base desde PHP.....	345
Cómo eliminar datos de una base con PHP.....	357
Cómo modificar datos de una base con PHP.....	365
Radiografía de un sistema con back-end y front-end.....	379

#### APÉNDICE

##### PROGRAMACIÓN ORIENTADA A

<b>OBJETOS</b> .....	387
----------------------	-----

#### APÉNDICE WEB

##### ADAPTANDO SOFTWARE LIBRE

<http://libroweb.alfaomega.com.mx>

## 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.

Solo 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 capacidad 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  
@firt



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.

España

<http://libros.itmaster.es> - [info@itmaster.es](mailto:info@itmaster.es)

México

<http://libros.itmaster.com.mx> - [info@itmaster.com.mx](mailto:info@itmaster.com.mx)

Argentina y otros países

<http://libros.itmaster.com.ar> - [info@itmaster.com.ar](mailto:info@itmaster.com.ar)

## Plataforma de contenidos interactivos

---

Para tener acceso al material de la plataforma de contenidos interactivos del libro: *PHP: creación de páginas Web dinámicas*, 2ª edición, siga los siguientes pasos:

1. Ir a la página: <http://libroweb.alfaomega.com.mx>
2. Ir a la sección *Catálogo* y seleccionar la imagen de la portada del libro, al dar doble clic sobre ella, tendrá acceso al material descargable.

NOTA: Se recomienda respaldar los archivos descargados de las páginas web en un soporte físico.

# Más allá de HTML y CSS

1

## ¡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.

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

**Tabla 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.

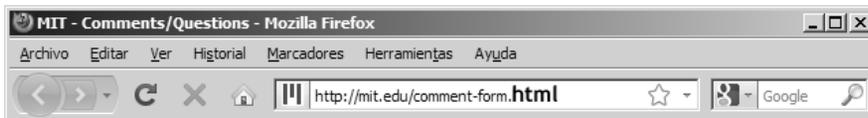
PHP es un acrónimo de *PHP: Hypertext Preprocessor*, es decir, “Preprocesador de Hipertexto marca PHP”. El hecho de que sea un preprocesador es lo que marca la diferencia entre el proceso que sufren las páginas Web programadas en PHP del de aquellas páginas Web comunes, escritas solo en lenguaje HTML.

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

## Proceso de archivos HTML

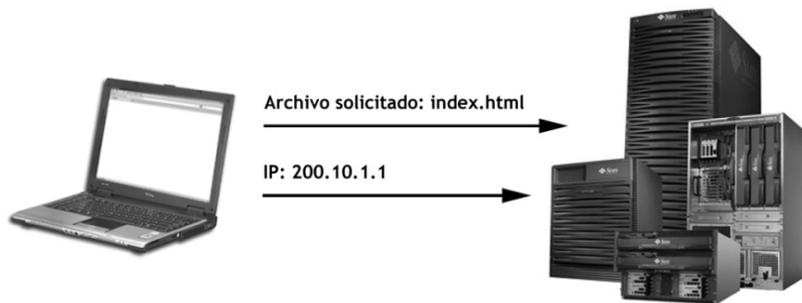
¿Cuál es el camino que sigue una página Web común (escrita en lenguaje HTML) desde que escribimos su dirección en nuestro navegador hasta que la vemos en nuestra pantalla?

- 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.**

- 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.
- 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.



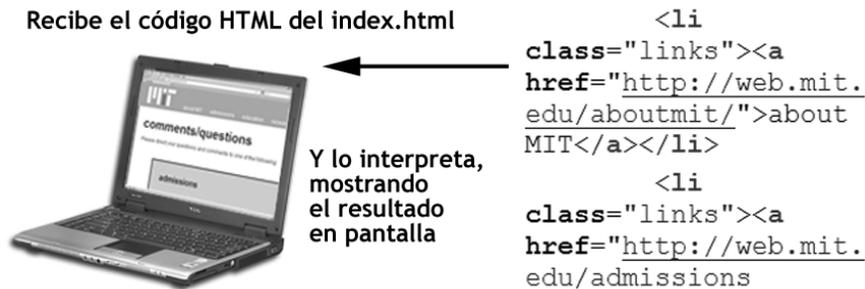
**Figura 1-3. El servidor Web busca en el disco rígido del *hosting* el archivo solicitado.**

- 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.
- 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.



**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.



**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 (*HyperText Transfer Protocol* o Protocolo de Transferencia de Hipertexto).

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

## Preproceso 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:

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



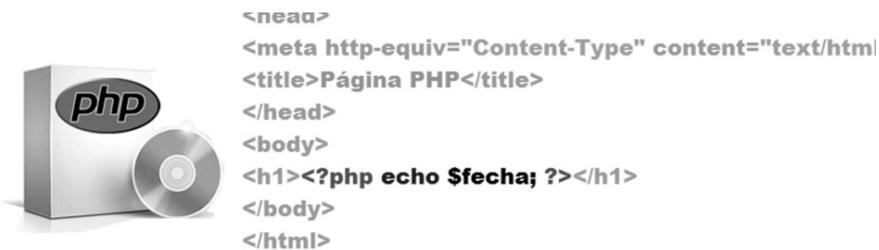
**Figura 1-6. URL de un archivo PHP.**

- 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.**

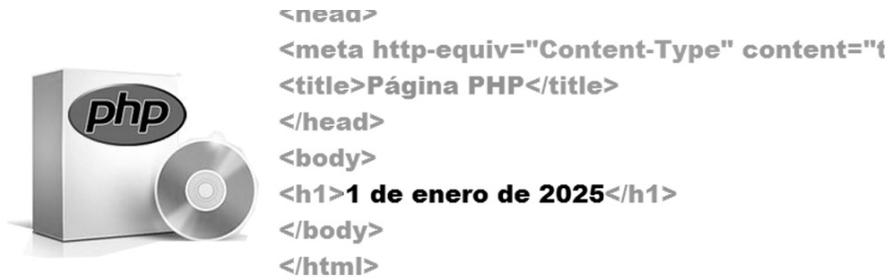
- Este programa intérprete de PHP busca en el disco rígido del *hosting* el archivo .php que fue solicitado, y comienza a leer su código línea por línea, buscando determinadas “marcas” o etiquetas que nosotros, como programadores, hemos dejado escritas y que contienen órdenes destinadas a ese programa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



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

- Cuando este programa intérprete de lenguaje PHP encuentra estas órdenes, las ejecuta (las procesa) y, a continuación, reemplaza todas las

órdenes que hubiese entre la apertura y el cierre de la etiqueta de PHP por el resultado de procesar esas órdenes. Es decir, **borra** las órdenes del código HTML en el que estaban escritas y, en su lugar, coloca los **datos** obtenidos como consecuencia de la ejecución de esas órdenes.



**Figura 1-9. El intérprete de PHP borra cada orden y en su lugar escribe el resultado de procesar esa orden.**

Veamos un ejemplo: supongamos que en una de esas órdenes le dijimos al intérprete de PHP que averigüe la fecha actual y que la escriba en el lugar exacto donde dejamos la orden escrita. Ésta se borrará del código fuente y, en su lugar, **quedará escrita la fecha**.

No se escribe así de simple y en español, pero ése es el concepto. Por esta razón, no debemos ejecutarlo porque no funcionará. Lo mencionamos, simplemente, para ilustrar la idea.

```

...
<body>
<h1>
<?php
Atención, software de PHP!
Quiero que escribas la fecha actual aquí!
?>
</h1>
</body>
...

```

- El software de PHP ejecuta la orden que le dejamos escrita y, al finalizar, devuelve al software servidor Web **el texto y el código HTML producido**, para que el servidor Web lo entregue **al navegador**, que lo interpreta como si este código HTML, que incluye la fecha, hubiese estado escrito allí desde un principio:

```
...  
<body>  
<h1>1 de enero de 2025</h1>  
</body>  
...
```

- En el código fuente que le llega al **navegador**, no vemos ningún rastro de la orden que habíamos escrito para el software de PHP, ya que este software se ocupó de borrarla para que nadie la vea, y en el lugar exacto en el que habíamos escrito esa orden, colocó “el resultado de ejecutar esa orden”, es decir, la fecha, que se ocupó de conseguir.

En resumen, el **preproceso** de páginas PHP consiste en esta serie de pasos: dejamos escritas entre medio de nuestras páginas algunas **órdenes** destinadas al **software intérprete de PHP** (órdenes que casi siempre consisten en que el software de PHP obtenga cierta información, como la fecha del ejemplo anterior); luego, colocamos otras órdenes para que el software intérprete de PHP “realice algo” con esa información, típicamente, que la **escriba** dentro del código fuente de la página HTML que se enviará al navegador del usuario.

En la Figura 1-10, observamos algunas órdenes en lenguaje PHP escritas en medio del código HTML de una página:

```
</nead>  
<body>  
<h1><?php echo $fecha; ?></h1>  
<h2><?php echo $titulo; ?></h2>  
<p><?php echo $noticia; ?></p>  
</body>  
</html>
```

**Figura 1-10.** Ejemplos de órdenes PHP intercaladas en el código HTML.

## Páginas estáticas

De la diferencia entre los procesos que sufren las páginas HTML comunes y las páginas PHP, podemos concluir que las páginas Web, escritas únicamente en lenguaje HTML, son **estáticas**: es decir, **nunca cambian** su contenido: pase lo que

pase, lo que llegará al navegador del usuario es lo que ha sido escrito en ellas por el diseñador Web, ni más ni menos, siempre lo mismo.

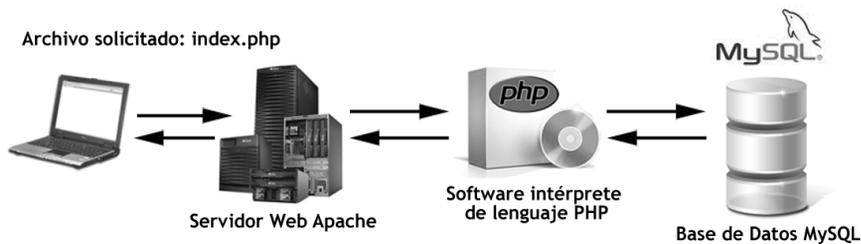
### Páginas dinámicas

Por el contrario, las páginas que incluyen código escrito en lenguaje PHP, nos dan la oportunidad de **personalizar** su contenido sobre la base de ciertas órdenes escritas (como en el ejemplo anterior de la fecha, la página hoy mostrará una fecha y mañana otra, y así sucesivamente; es decir: siempre generará un **contenido distinto**, variable).

El contenido de esas páginas, al menos en partes de ellas, cambiará y no será siempre el mismo, ya que dependerá de la información que obtenga el software de PHP y coloque en ellas. Serán páginas dinámicas.

### Las bases de datos

El concepto de páginas dinámicas que acabamos de esbozar, se complementa a la perfección con las **bases de datos**, ya que éstas se ocupan de almacenar datos y, las páginas dinámicas, de **leerlos** y **mostrarlos** dentro de ellas.



**Figura 1-11. Esquema de interacción entre los programas que participan cuando las páginas utilizan una base de datos.**

### MySQL: la base de los proyectos Web exitosos

Desde la aparición de PHP, la base de datos que siempre estuvo asociada a PHP fue MySQL, no solo gracias a su gran potencia y rapidez, sino, fundamentalmente, a que su licencia no tenía costo para los *hostings* (a diferencia de Oracle, SQL Server y otros competidores), lo que contribuyó a su rápida difusión y a su mejoramiento vertiginoso en sucesivas versiones, que contribuyeron a que se convirtiera en la base de datos que provee almacenamiento a sistemas de la envergadura de Google, Facebook y millones de otros populares sitios Web en todo el mundo.

Pero, ¿para qué sirve usar una base de datos en un sitio Web? Veamos algunos de sus posibles usos.

## Un camino de ida: del almacén a las páginas

Pensemos en un ejemplo: una página que muestra un catálogo de libros:



Figura 1-12. Página que muestra información de una base de datos.

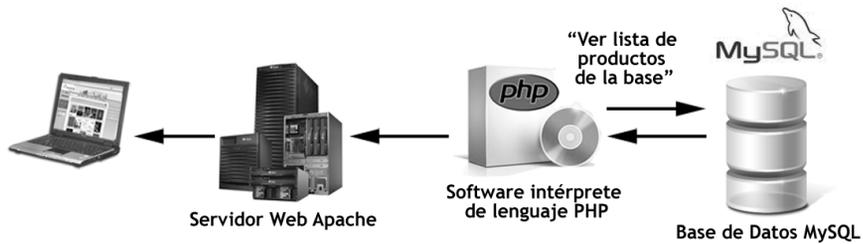
Si pretendiéramos hacer este tipo de sitios únicamente con HTML, deberíamos crear esta página manualmente y, en caso de alguna modificación en los datos de alguno de los productos (un nuevo producto, o quitar uno existente), deberíamos modificar el archivo HTML a mano, y volver a colocar en el *hosting* una copia actualizada de esta página HTML mediante FTP. Así, en todos los cambios de cualquiera de los miles de artículos que se muestran.

En cambio, con PHP y MySQL, podemos crear una **base de datos** que contenga información sobre los miles de productos que se mostrarán. Luego, utilizaremos una **página dinámica** escrita en PHP, que incluya la orden de leer esa base de datos y publicar los productos. Si modificamos los datos, éstos se reemplazarán directamente en la base de datos, sin necesidad de modificar ni una línea de las páginas Web que exhiben los productos, ya que la orden sigue siendo exactamente la misma: “leer la base y mostrar todos los productos”.

Otra vez, para graficar la idea, la simplificaremos, pero pronto lo aprenderemos a programar para que funcione:

```

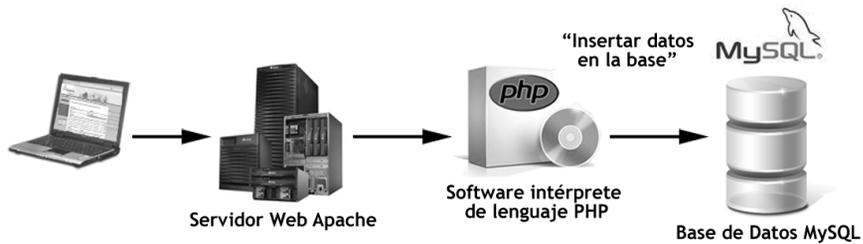
...
<p>
<?php
Atención, software de PHP!:
Quiero que le pidas a la base de datos la lista de
productos y la muestres aquí!
?>
</p>
...
    
```



**Figura 1-13. Esquema del circuito de una página que muestra información almacenada en una base de datos.**

**Un camino de vuelta: recepción de datos de formularios**

Pero aquí no terminan las ventajas de almacenar en una base de datos la información de una página, sino que PHP y MySQL agregan un camino imposible de lograr solamente con HTML: la posibilidad de enviar datos desde el navegador del usuario hacia el servidor, y que estos datos queden almacenados en la base de datos del *hosting*.



**Figura 1-14. Esquema del circuito de una página que envía datos hacia el servidor.**

¿Posibles usos?: envío de formularios con consultas, currículums, formularios para paneles de administración que actualizan datos de catálogos, envío de mensajes para publicar en foros, envío de fotografías, etc.

The screenshot shows a web application interface for Ediciones Paidós. The main content area is a form for editing book data. At the top, there is a search bar with the text "¿COMO BUSCAR?" and radio buttons for "AUTOR", "TITULO", "ISBN", and "CODIGO". Below the search bar, there is a navigation menu with categories like "HUMANIDADES", "PSICOLOGIA", "PSICOMETRIA", "PSICANALISIS", "EDUCACION", "DIVULGACION", and "EMPRESA". The form itself has a header that says "Ud. puede modificar todos los datos de este Libro, salvo su código, y salvo su tema". The form fields are:
 

- Código: 10036
- Título: SIMBIOSIS Y AMBIGÜEDAD
- Subtítulo: Estudio psicoanalítico
- Autor: Bleger, José
- Tema: Colección: 2 PSICOLOGÍA PROFUNDA
- Descripción: Este libro parte de la hipótesis de que el núcleo interdependencia simbiótica es de carácter amb relación de la simbiosis con el proceso de pro y las modalidades de la relación objetal, y se estructura de la ambigüedad en la clínica. A t la ambigüedad. se reconsideran conceptos como
- Subgénero: Teoría y clínica
- Público: Universitario y Profesional
- ISBN: 950-12-4038-3
- Medida: 13,5 x 22
- Páginas: 320
- Código de Barras: 9 789501 240388
- Mes: Diciembre
- Año: 2003

 There is a "Modificar" button at the bottom of the form and a link to "- Volver al Panel -".

**Figura 1-15.** Ejemplo de formulario que envía datos al servidor y los guarda en una base de datos.

### **Sitios autoadministrables: CMS con Front-end y Back-end**

La combinación de los dos “caminos” anteriores (enviar datos **hacia** la base de datos y, luego, **mostrar** esos datos mediante páginas dinámicas) permiten crear sistemas denominados CMS (*Content Management System* o Sistemas de Administración de Contenidos) que gestionan la publicación de contenidos de portales, sistemas de comercio electrónico, comunidades *on-line*, y muchas otras aplicaciones, siempre sobre la base de esos conceptos básicos: el usuario envía datos hacia el *hosting*, que se almacenan en la base de datos y, luego, esos datos se muestran (a él o a otros usuarios autorizados).

Se denomina *Front-end* a la parte visible por los usuarios, y *Back-end* al panel de administración de acceso restringido solo a los usuarios autorizados (como el de la última imagen).

### **Millares de CMS de uso libre y gratuito**

Existen literalmente miles de CMS con distintas finalidades, listos para usar, con licencia de libre uso y modificación (típicamente, la licencia GPL y similares), creados con PHP y MySQL. Algunos de los más famosos son *Joomla*, *WordPress*, *osCommerce*, *Mambo*, *Drupal*, *Moodle*, *Magento*, *OpenRealty*, pero existen miles más.

El secreto del negocio está en **saber programar en PHP** lo suficiente como para **entender** y poder realizar algunas **modificaciones** para adaptar estos CMS a las necesidades específicas de cada cliente, y poder ofrecerles soluciones de bajo costo y muy rápida implementación.



**Figura 1-16. Nuestro negocio: tomar código prearmado y modificarlo según las necesidades de nuestros clientes.**

De aprender lo necesario de PHP y MySQL para poder hacer esas modificaciones, es de lo que nos ocuparemos en el resto de este libro.