LENGUAJE PHP

Amanda Collado Blanco

Department of Computer Science and Automation, University of Salamanca

Plaza de la Merced, s/n, 37008, Salamanca, Spain

acolladoblanco@usal.es

1. Introducción



**Figura 1** Logo PHP

PHP, acrónimo recursivo en inglés de PHP: Hypertext Preprocessor (preprocesador de hipertexto), es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el HTML resultante.

PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en muchos sistemas operativos y plataformas sin ningún costo.

Fue creado originalmente por Rasmus Lerdorf en el año 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. ​ Este lenguaje forma parte del software libre publicado bajo la licencia PHPv3\_01, es una licencia Open Source validada por Open Source Initiative. La licencia de PHP es del estilo de licencias BSD, esta licencia no tiene restricciones de copyleft" asociadas con GPL.

PHP puede ser desplegado en la mayoría de los servidores web y en todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. El número de sitios basados en PHP se ha visto reducido progresivamente en los últimos años, con la aparición de nuevas tecnologías como Node.JS, Golang, ASP.NET, etc. El sitio web de Wikipedia está desarrollado en PHP. ​ Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Aunque todo en su diseño está orientado a facilitar la creación de sitios webs, es posible crear aplicaciones con una interfaz gráfica de usuario, utilizando alguna extensión como puede ser PHP-Qt, PHP-GTK,6​ WxPHP, WinBinder, Roadsend PHP, Phalanger, Phc o HiP Hop VM. También puede ser usado desde la línea de comandos, de la misma manera como Perl o Python pueden hacerlo; a esta versión de PHP se la llama PHP-CLI (Command Line Interface). ​

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo, obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente.

Mediante extensiones es también posible la generación de archivos PDF,8​ Flash, así como imágenes en diferentes formatos.

Permite la conexión a diferentes tipos de servidores de bases de datos tanto SQL como NoSQL tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird, SQLite o MongoDB.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como Unix (y de ese tipo, como Linux o Mac OS X) y Microsoft Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

PHP es una alternativa a las tecnologías de Microsoft ASP y ASP.NET (que utiliza C# y Visual Basic .NET como lenguajes), a ColdFusion de la empresa Adobe, a JSP/Java, CGI/Perl y a Node.js/Javascript. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un entorno de desarrollo integrado comercial llamado Zend Studio. CodeGear (la división de lenguajes de programación de Borland) ha sacado al mercado un entorno de desarrollo integrado para PHP, denominado 'Delphi for PHP. También existen al menos un par de módulos para Eclipse, uno de los entornos más populares.

PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc.

2. Historia

Fue originalmente diseñado en Perl, con base en la escritura de un grupo de CGI binarios escritos en el lenguaje C por el programador danés-canadiense Rasmus Lerdorf en el año 1994 para mostrar su currículum vítae y guardar ciertos datos, como la cantidad de tráfico que su página web recibía. El 8 de junio de 1995 fue publicado "Personal Home Page Tools" después de que Lerdorf lo combinara con su propio Form Interpreter para crear PHP/FI.

Dos programadores israelíes del Technion, Zeev Suraski y Andi Gutmans, reescribieron el analizador sintáctico (parser, en inglés) en 1997 y crearon la base del PHP3, y cambiaron el nombre del lenguaje por PHP: Hypertext Preprocessor.2​ Inmediatamente comenzaron experimentaciones públicas de PHP3, y se publicó oficialmente en junio de 1998. Para 1999, Suraski y Gutmans reescribieron el código de PHP, y produjeron lo que hoy se conoce como motor Zend. También fundaron Zend Technologies en Ramat Gan, Israel.2​

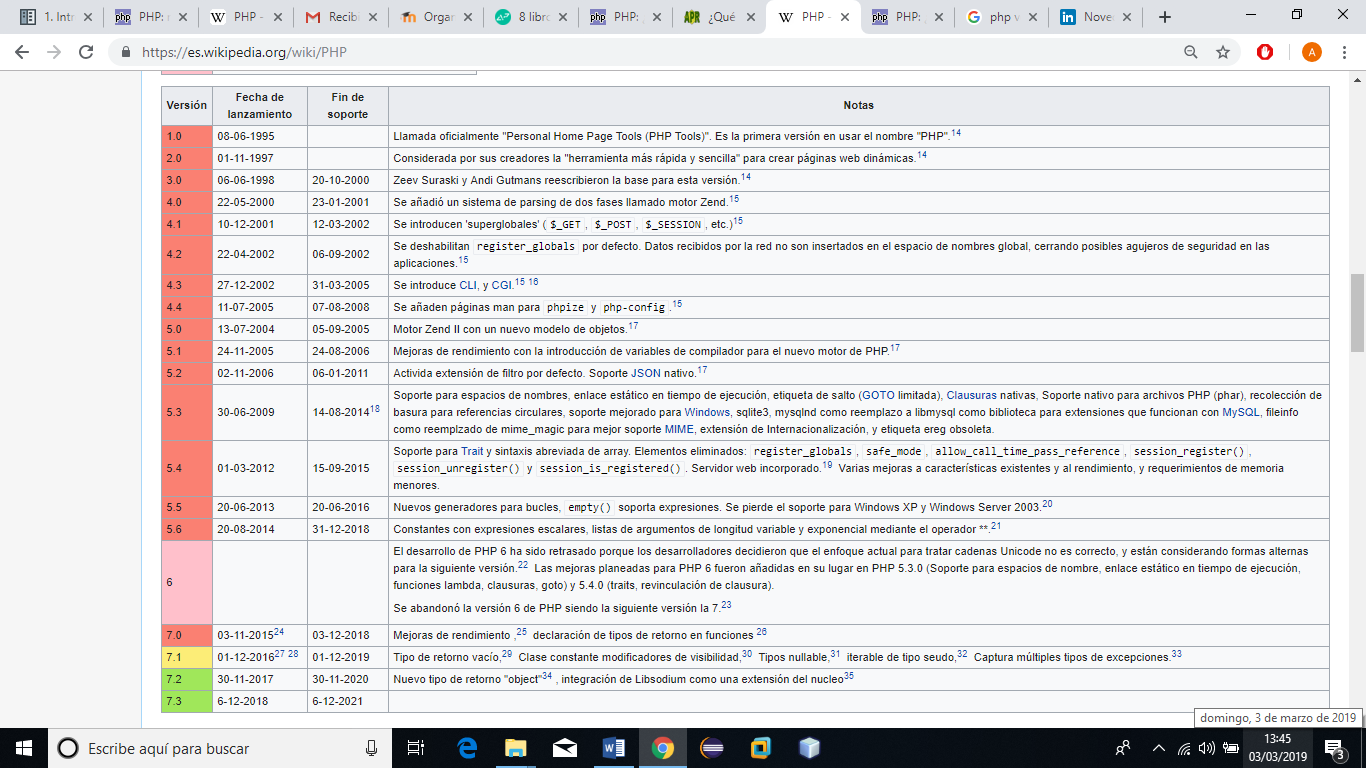
En mayo del 2000, PHP 4 se lanzó bajo el poder del motor Zend 1.0. El 13 de julio del 2007 se anunció la suspensión del soporte y desarrollo de la versión 4 de PHP,11​ y, a pesar de lo anunciado, se ha liberado una nueva versión con mejoras de seguridad, la 4.4.8, publicada el 13 de enero del 2008, y posteriormente la versión 4.4.9, publicada el 7 de agosto del 2008.12​ Según esta noticia,13​ se le dio soporte a fallos críticos hasta el 9 de agosto del 2008.

El 13 de julio del 2004, se lanzó PHP 5, utilizando el motor Zend Engine 2.0 (o Zend Engine 2).2​ Incluye todas las ventajas que provee el nuevo Zend Engine 2.

El 3 de noviembre del 2015, se lanzó PHP 7, incluyendo las siguientes majoras:

* **Rendimiento y velocidad.** El núcleo mejorado de Zend Engine 3.0, un motor de ejecución que interpreta PHP, repercute en una mayor velocidad y una mejora en el rendimiento bastante significativa. La propia Zend Technologies estima que, por ejemplo en Magento, en uso de memoria se reduce hasta en un 30%, por lo que se puede ejecutar en la mitad de tiempo, respeto a la versión anterior PHP 5.6. Otros ejemplos son Drupal 8, que se ejecuta un 72% más rápido o WordPress que tan solo consume el 25% de la CPU para su funcionamiento.
* **Funcionalidad.** Cada nueva versión de PHP incorpora nuevas funcionalidades y PHP 7 no iba a ser menos. Estas nuevas incorporaciones permiten tener un código más limpio, detectar errores más fácilmente y programar con mayor eficiencia y fiabilidad.
* **Mayor espacio.** Las funcionalidades obsoletas se eliminan lo que permite liberar espacio para permitir optimizaciones.
* **Declaraciones de tipo precisas.** Integra 4 declaraciones de tipo escalar: int (entero), bool (booleano), float (decimal) y string (cadena).

El 6 de diciembre del 2018, se lanzó PHP 7.3, que a día de hoy es la última versión y por tanto la que se está utilizando.



**Figura 2** Versiones PHP

3. Funcionamiento

PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa CGI, como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies. Aunque PHP puede hacer mucho más.

Existen principalmente tres campos principales donde se usan scripts de PHP:

* **Scripts del lado del servidor**. Este es el campo más tradicional y el foco principal. Son necesarias tres cosas para que esto funcione. El analizador de PHP (módulo CGI o servidor), un servidor web y un navegador web. Es necesario ejecutar el servidor con una instalación de PHP conectada. Se puede acceder al resultado del programa de PHP con un navegador, viendo la página de PHP a través del servidor. Todo esto se puede ejecutar en su máquina si está experimentado con la programación de PHP. Véase la sección sobre las instrucciones de instalación para más información.
* **Scripts desde la línea de comandos**. Se puede crear un script de PHP y ejecutarlo sin necesidad de un servidor o navegador. Solamente es necesario el analizador de PHP para utilizarlo de esta manera. Este tipo de uso es ideal para scripts que se ejecuten con regularidad empleando cron (en \*nix o Linux) o el Planificador de tareas (en Windows). Estos scripts también pueden usarse para tareas simples de procesamiento de texto. Véase la sección Uso de PHP en la línea de comandos para más información.
* **Escribir aplicaciones de escritorio**. Probablemente PHP no sea el lenguaje más apropiado para crear aplicaciones de escritorio con una interfaz gráfica de usuario, pero si se conoce bien PHP, y se quisiera utilizar algunas características avanzadas de PHP en aplicaciones del lado del cliente, se puede utilizar PHP-GTK para escribir dichos programas. También es posible de esta manera escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal.

PHP puede emplearse en todos los sistemas operativos principales, incluyendo Linux, muchas variantes de Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente otros más. PHP admite la mayoría de los servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros.

De modo que, con PHP, se tiene la libertad de elegir el sistema operativo y el servidor web. Además, se tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos (POO), o una mezcla de ambas.

Con PHP no se está limitado a generar HTML. Entre las capacidades de PHP se incluyen la creación de imágenes, ficheros PDF e incluso películas Flash (usando libswf y Ming) generadas sobre la marcha. También se puede generar fácilmente cualquier tipo de texto, como XHTML y cualquier otro tipo de fichero XML. PHP puede autogenerar estos ficheros y guardarlos en el sistema de ficheros en vez de imprimirlos en pantalla, creando una caché en el lado del servidor para contenido dinámico.

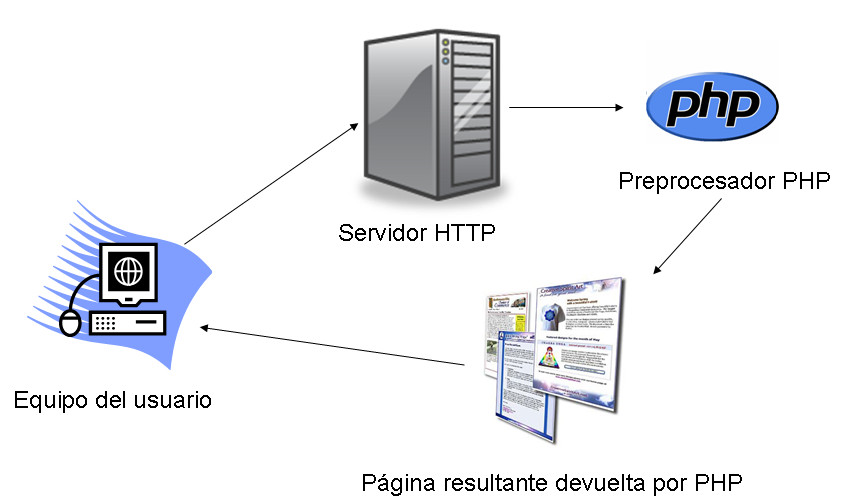
Una de las características más potentes y destacables de PHP es su soporte para un amplio abanico de bases de datos. Escribir una página web con acceso a una base de datos es increíblemente simple utilizando una de las extensiones específicas de bases de datos (p.ej., para mysql), o utilizar una capa de abstracción como PDO, o conectarse a cualquier base de datos que admita el estándar de Conexión Abierta a Bases de Datos por medio de la extensión ODBC. Otras bases de datos podrían utilizar cURL o sockets, como lo hace CouchDB.

PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. También se pueden crear sockets de red puros e interactuar usando cualquier otro protocolo. PHP tiene soporte para el intercambio de datos complejos de WDDX entre virtualmente todos los lenguajes de programación web.

El lenguaje PHP se procesa en servidores, que son potentes ordenadores con un software y hardware especial. Cuando se escribe una dirección en un navegador web como Internet Explorer, Firefox o Chrome, ¿qué ocurre? Se envían los datos de la solicitud al servidor que los procesa, reúne los datos (por eso decimos que es un proceso dinámico) y el servidor lo que devuelve es una página HTML como si fuera estática.

El esquema es (**Figura 3**):

1. Petición de página web al servidor
2. El servidor recibe la petición, reúne la información necesaria consultando a bases de datos o a otras páginas webs, otros servidores, etc.
3. El servidor responde enviando una página web “normal” (estática) pero cuya creación ha sido dinámica (realizando procesos de modo que la página web devuelta no siempre es igual).



**Figura 3** Funcionamiento de PHP

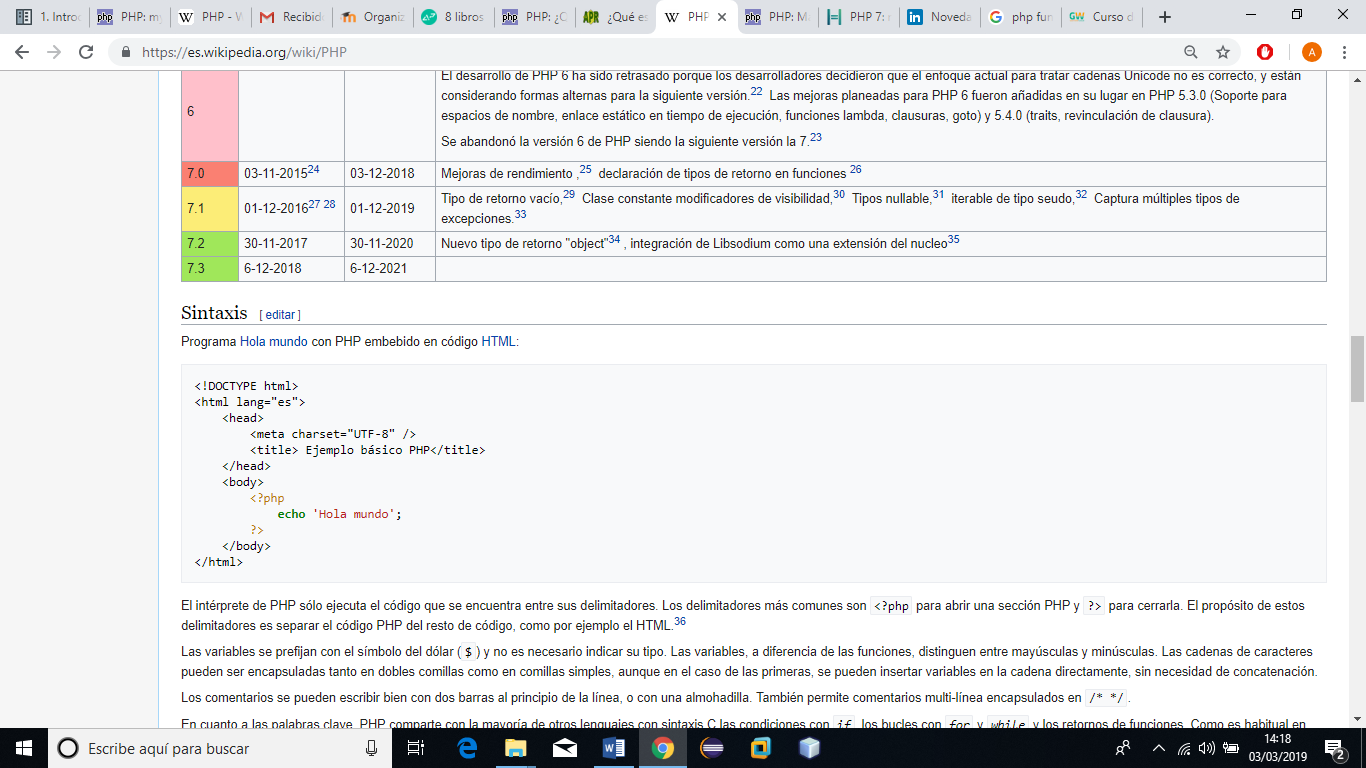
4. Sintaxis

El intérprete de PHP sólo ejecuta el código que se encuentra entre sus delimitadores. Los delimitadores más comunes son ***<?php*** para abrir una sección PHP y ***?>*** para cerrarla. El propósito de estos delimitadores es separar el código PHP del resto de código, como por ejemplo el HTML.

Las variables se prefijan con el símbolo del dólar **($)** y no es necesario indicar su tipo. Las variables, a diferencia de las funciones, distinguen entre mayúsculas y minúsculas. Las cadenas de caracteres pueden ser encapsuladas tanto en dobles comillas como en comillas simples, aunque en el caso de las primeras, se pueden insertar variables en la cadena directamente, sin necesidad de concatenación.

Los comentarios se pueden escribir bien con dos barras al principio de la línea, o con una almohadilla. También permite comentarios multi-línea encapsulados en /\* \*/.

En cuanto a las palabras clave, PHP comparte con la mayoría de otros lenguajes con sintaxis C las condiciones con **if**, los bucles con **for** y **while** y los retornos de funciones. Como es habitual en este tipo de lenguajes, las sentencias deben acabar con punto y coma **(;)**.



**Figura 4** Hola Mundo con PHP embebido en HTML

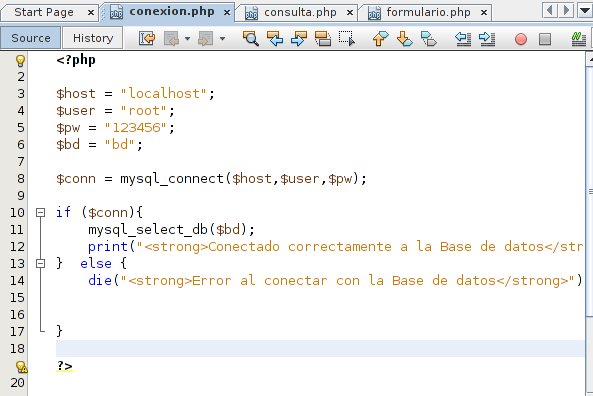
5. PHP y MySQL

PHP puede combinarse con MySQL para trabajar con bases de datos. En la actualidad esta última se utiliza muchísimo.

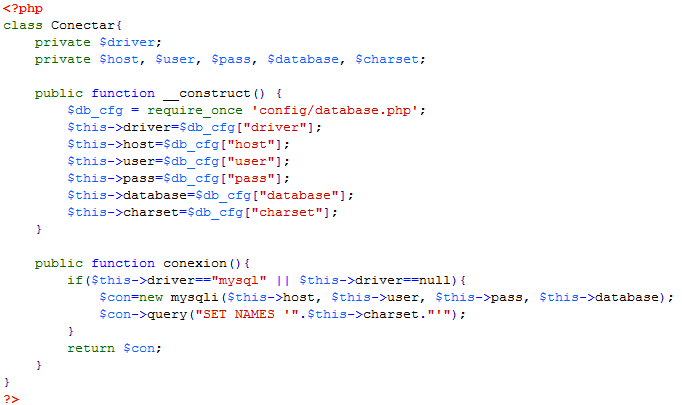
Existen más, pero las tres formas más populares de conectarse desde PHP a una base de datos de MySQL: mysql, mysqli y PDO (PHP Data Objects). En cada proyecto se puede usar una manera u otra, pero para saber cuál es la más recomendada, hay que conocer en que se distinguen:

* **mysql** es la API histórica que ofrece PHP. A partir de la versión PHP 5 se utiliza mysqli. **Figura 5.**
* **mysqli** es la nueva versión de mysql, orientada a objetos que se utiliza acualmente. **Figura 6**.
* **PDO** es una capa de abstracción de base de datos, con soporte para MySQL y otros sistemas gestores de bases de datos. Proporciona comandos preparados y otra serie de operaciones adaptadas al driver necesario. **Figura 7.**

Si no hubiera ningún motivo que hiciera variar la elección, se recomienda usar PDO; y si no es posible, entonces mysqli. Como último recurso, usar mysql.



**Figura 5** Ejemplo de conexión utilizando MySQL y PHP sin el estilo orientado a objetos.



**Figura 6** Ejemplo de inicio de sesión con usuario y contraseña utilizando MySQL y PHP con el estilo orientado a objetos.



**Figura 7** Conexión a MySQL utilizando PDO en PHP

6. Características

* Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
* Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de php arrays.
* El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.
* Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
* Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
* Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
* Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
* Permite aplicar técnicas de programación orientada a objetos.
* No requiere definición de tipos de variables, aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
* Tiene manejo de excepciones (desde PHP5).
* Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador​ puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.
* Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones WEB de manejo de contenido, y es su uso principal.

Referencias

1. Wikipedia, PHP, https://es.wikipedia.org/wiki/PHP (2019).

2. PHP.NET - Manual de PHP, http://php.net/manual/es/index.php (2019).

3. HostEurope blog, PHP 7 - novedades y mejoras, https://www.hosteurope.es/blog/novedades-y-mejoras-de-php-7 (2019).

4. Aprender a programar, ¿Qué es PHP? y ¿para qué sirve? Un potente lenguaje de programación para crear páginas web. https://www.aprenderaprogramar.com/index.php?option=com\_content&view=article&id=492&catid=70&Itemid=193 (2019).

5. Gestiona tu web, curso de iniciación a PHP, https://www.gestionatuweb.net/curso-de-php/ (2019).

6. Baul PHP, PDO conectar con la base de datos MySQL , https://www.baulphp.com/pdo-conectar-con-la-base-de-datos-mysql/ (2019).

7. Hostinger, Cómo conectar PHP con MySQL, https://www.hostinger.es/tutoriales/conectar-php-mysql/ (2019).