

Metatitle: Coderhouse | Aprende que es y como programar con PHP

Metadescripción: Todo sobre el lenguaje de programación PHP En el artículo aprenderás no solo sobre sus diferentes versiones, sino también cómo funciona y mucho más. ¿Qué esperas para leerlo?

¿Qué es PHP y para qué sirve? Usos, características y cómo aprender

Todo sobre este lenguaje de programación del lado del servidor. Su origen, la evolución de sus versiones y las ventajas a la hora de crear sitios web.

Tabla de contenido

- [¿Qué es y para qué sirve PHP?](#)
- [¿Qué significa PHP?](#)
- [Origen y evolución de PHP](#)
- [¿Cuáles son las versiones de PHP?](#)
- [Ejemplo de código PHP](#)
- [¿Cómo funciona PHP?](#)
- [¿En qué plataformas corre PHP?](#)
- [¿En qué situaciones se recomienda PHP?](#)
- [Ventajas y desventajas de PHP](#)
- [¿Qué se necesita para desarrollar en PHP?](#)

¿Qué es y para qué sirve PHP?

PHP es un lenguaje de programación de código abierto. En concreto, este lenguaje de scripting es uno de los más populares entre los [desarrolladores backend](#) (del lado del servidor).

Se emplea para la creación de sitios web. Además, es el motor de los CMS más utilizados, entre ellos WordPress y Joomla.

De acuerdo al [manual oficial de PHP](#), presenta las siguientes características destacadas:

- Puede ser embebido en HTML.
- Una parte importante de su sintaxis proviene de Java, C y Perl, con el añadido de funciones propias.
- El objetivo del **lenguaje PHP** es permitir a los [programadores backend](#) escribir con rapidez páginas que se generan en forma dinámica.
- Además, ofrece un alto grado de compatibilidad con otras bases de datos.

¿Qué significa PHP?

PHP significa “**PHP: Hypertext Preprocessor**”.

Se trata de un acrónimo complejo, del tipo recursivo: son aquellos que no parecen explicar el significado completo. Por ejemplo, HTML se refiere a “HyperText Markup Language”, en el que encontramos todas las letras.

El **significado de PHP** puede resultar complejo porque la primera palabra es el acrónimo mismo.



Imagen: <https://unsplash.com/es/fotos/2Rd-hwT2xQ0>

Origen y evolución de PHP

El **origen del lenguaje PHP** se remonta a mediados de la década del 90', cuando el programador groenlandés [Rasmus Lerdorf](#) creó un sistema para interpretar y procesar formularios denominado PHP/FI.

Cuando se presentó, ese conjunto de *scripts* apareció con el nombre “Personal Home Page”. A esa versión se la bautizó con el acrónimo PHP.

En 1995, Lerdorf liberó el código fuente para que otros programadores puedan usarlo. El paso a código abierto expandió la **comunidad PHP**, que aprovechó las ventajas del lenguaje para la simplificación de tareas de programación.

La primera versión con mejoras relevantes fue **PHP 3**, que comenzó a parecerse al lenguaje que hoy circula. Fue desarrollada por la comunidad de *developers* y compatible con todos los sistemas operativos más populares.

PHP 3 se postuló como una herramienta útil para los [desarrolladores backend](#), aquellos que trabajan del lado del servidor, es decir, en el “detrás de escena” de los sitios web y aplicaciones.

Se destacó por su uso intuitivo y por las funciones integradas en su núcleo que permitían gestionar bases de datos, sistemas de archivos, generar imágenes dinámicas, etcétera.

A partir de PHP, el lenguaje cambia su nombre y toma el acrónimo recursivo antes señalado: **PHP: Hypertext Preprocessor**.

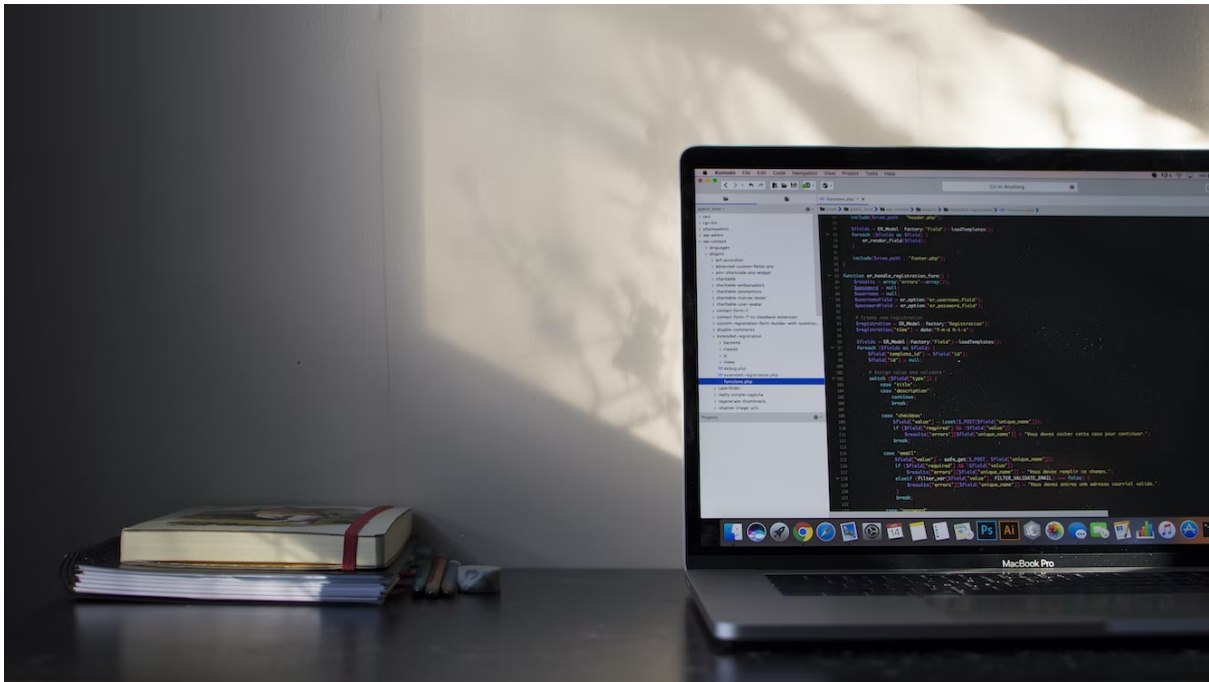
Le siguieron un buen número de **versiones de PHP**, que repasamos en el siguiente punto.

¿Cuáles son las versiones de PHP?

Con el desarrollo inicial de Rasmus Lerdorf y la irrupción de PHP 3 una vez que el código se abrió a la comunidad, este lenguaje de scripting evolucionó a través de una serie de versiones.

En lo que sigue, repasamos las características de las **versiones de PHP** más relevantes:

- **PHP 4.** Se lanzó en el año 2000 con una reescritura completa del núcleo PHP, que aumentó el rendimiento y facilitó el mantenimiento del código. Si bien sumaba opciones valiosas (como el soporte para múltiples servidores) y por eso ganaba popularidad, aún era un lenguaje inmaduro en comparación con otros que llevaban un recorrido más extenso.
- **PHP 5.** De 2004, se considera a esta versión como el punto de madurez del lenguaje. Se incluyeron mejoras que permitieron el desarrollo de aplicaciones más grandes. Las iteraciones de PHP 5 sumaron características relevantes, como los “namespaces” y los “traits”. PHP comenzó a codearse con las herramientas más avanzadas en el sector como Java o Ruby.
- **Composer.** Junto con las mejoras de la versión PHP 5, llegó una de las herramientas más populares en este ecosistema. La referencia es a Composer, su gestor de dependencias.
- **PHP 7.** La llegada de esta versión en el año 2015 trajo importantes mejoras en este lenguaje de programación. Duplicó el rendimiento de las aplicaciones y en ediciones posteriores (PHP 7.3) llegó a triplicar la velocidad de PHP 5.
- **PHP 8.** Aparecida en 2020, es la generación actual; emplea el motor Zen 4 que ofrece características adicionales de POO (Programación Orientada a Objetos).



Imágen: <https://unsplash.com/es/fotos/xrVDYZRGdw4>

Ejemplo de código PHP

El siguiente es un **ejemplo de código PHP** introductorio, tomado del sitio php.net.

```
<!DOCTYPE html>

<html>

    <head>

        <title>Ejemplo</title>

    </head>

    <body>

        <?php

            echo "<strong>¡Hola, soy un script de PHP!</strong>";

        ?>

    </body>

</html>
```

¿Cómo funciona PHP?

El ejemplo de código nos permite comprender **cómo funciona PHP**.

Para comenzar, es preciso subrayar algunas de las características arriba señaladas:

- PHP es un lenguaje de programación backend.
- Es de código abierto.
- Por eso está en constante evolución y perfeccionamiento.

¿Cómo funciona, en la práctica? Mira.

En lugar de utilizar numerosos comandos para mostrar HTML, las **páginas de PHP** tienen HTML con código incrustado, a diferencia de lo que pasa en C o en Perl.

El [código de PHP](#) está encerrado entre las etiquetas especiales (de comienzo y final). Tal como nota en php.net, eso permite entrar y salir del modo PHP.

Diferencia entre PHP y JavaScript

Si bien PHP y JavaScript son lenguajes de programación con gran reconocimiento a nivel mundial, quienes trabajan en esto cuentan con cierta preferencia entre unos y otros.

La programación PHP, como decíamos anteriormente, es un lenguaje scripting pero del lado del servidor. Esto quiere decir, que se lleva a cabo desde el servidor web.

Asimismo, su funcionamiento parte de un contenido dinámico, el cual por lo general es una base de datos programada en una web para por ejemplo brindar un mensaje de bienvenida cuando el usuario ingresa a la misma.

En cambio, el lenguaje JavaScript al encontrarse del lado del cliente, su desarrollo va a variar de acuerdo al dispositivo que use el usuario en ese momento.

Un claro ejemplo es un plugin de preguntas frecuentes, en el cual es el JavaScript del lado del cliente en acción. Al hacer clic o tocar una pregunta, los manejadores de eventos de JavaScript activan o desactivan las propiedades de visibilidad o visualización del CSS, mostrando u ocultando la respuesta relevante.

¿Qué ocurre del lado del navegador o del cliente? Se recibe el resultado del script, sin que se conozca el código subyacente.

Para retomar, la programación PHP es creada con el objetivo en el desarrollo de sitios web y aplicaciones, **PHP favorece la conexión entre los servidores y las interfaces de usuario**.

Por lo demás, uno de los usos principales del **lenguaje PHP** es estructurar sitios web en plataformas CMS, entre ellas WordPress.

Finalmente, la **API de funciones de PHP** es uno de sus puntos fuertes. En su núcleo hay funciones que permiten trabajar con una variedad de recursos y operaciones. Hay funciones para hacerlo con las bases de datos más populares, establecer conexiones por medio de HTTP o FTP, tratamiento de imágenes, envíos de correos, creación de PDF, sistema de archivos, etcétera.



Imagen: https://unsplash.com/es/fotos/R_W_9D-53lw

¿En qué plataformas corre PHP?

PHP es multiplataforma. Está disponible para los siguientes sistemas operativos:

- Linux.
- Windows.
- Mac.

Para **desarrollar en PHP** es preciso instalar todos los programas necesarios: servidor web, PHP y una base de datos.

Cabe señalar que las posibilidades varían de acuerdo a la computadora que se use para trabajar, al sistema operativo y a las singularidades de los desarrollos.

Por otra parte, **¿cuáles son las plataformas para el despliegue de aplicaciones y sitios creados con PHP?** La plataforma más frecuente es Linux.

Para la publicación de un proyecto, es posible contratar un servicio de hosting con soporte para PHP. También es habitual la contratación de un servidor propio (VPS, cloud o dedicado).

¿En qué situaciones se recomienda PHP?

El **principal uso de PHP** es para la creación de sitios web. Como hemos dicho, se destaca su capacidad para vincular el servidor con la interfaz de usuario, basándose en todo el código HTML.

Algunos de los **usos de PHP** más populares son:

- **Aplicaciones para sitios web.** Provee agilidad en los tiempos de respuesta y conexión a grandes bases de datos. PHP permite desarrollar sitios con agilidad, que tienen un gran rendimiento, incluso si están colmados de recursos. Para tener una dimensión de esto, un botón sirve de muestra. ¿Sabías que **Facebook usa PHP**?
- **Plataformas de e-commerce.** La necesidad de estos entornos de generar una comunicación constante con bases de datos de gran tamaño y complejas hace que el PHP sea una herramienta ideal, también en estos casos. En rigor, muchas plataformas de comercio electrónico programan su backend en PHP.
- **El caso de WordPress.** Este popular CMS basa la programación de sus aplicaciones en PHP. Además, este lenguaje aplica a los conocidos plugins de WordPress, que son sus herramientas adicionales, que ayudan a completar los proyectos web de un modo rápido y sencillo.

¿Qué se puede hacer sabiendo programación PHP?

Al contar con conocimientos en programación PHP tendrás la posibilidad de hacer cualquier cosa y como ya dijimos, PHP se enfoca en la programación de scripts del lado del servidor.

Es por eso que te contaremos los pilares en los cuales se basa la programación script:

1. **Scripts del lado del servidor.** Este es el campo más tradicional y el foco principal. Para su correcto funcionamiento se necesita de un 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.
2. **Scripts desde la línea de comandos.** Podrás crear un script de PHP y ejecutarlo sin necesidad de un servidor o navegador. Solamente vas a necesitar el analizador de PHP para utilizarlo de esta manera. Este tipo de uso es ideal para scripts que se ejecuten de forma regular empleando cron (en *nix o Linux) o el Planificador de tareas en el caso de Windows. Estos scripts también pueden usarse para tareas simples de procesamiento de texto.
3. Si bien PHP no se suele recomendar para esto, **permite escribir aplicaciones de escritorio gracias a php-gtk.**

Dado que PHP es compatible con diversos sistemas operativos como Windows, Linux, MacOS entre otros, podrás elegir desde cuál trabajar con la libertad de elegir el sistema operativo y el servidor web.

Que tengas conocimientos en programación PHP no significa que únicamente podrás limitarte a la creación de HTML. Si no que también vas a ser capaz de crear imágenes, PDF e incluso películas.

Asimismo, te permite escribir una página web haciendo hincapié en una base de datos, lo cual es una de las grandes características del lenguaje PHP.

El lenguaje de programación PHP tiene permitido el intercambio de datos de WDDX con otros lenguajes de programación mediante el uso de protocolos como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM en el caso de Windows y muchos otros.

PHP posee útiles características de procesamiento de texto, las cuales abarcan las expresiones regulares compatibles con Perl (PCRE), sumado a muchas extensiones y herramientas para contar con el acceso y analizar documentos XML.

Además, este lenguaje estandariza todas las extensiones XML sobre el fundamento sólido de libxml2, y amplía este conjunto de características al agregarle soporte para SimpleXML, XMLReader y XMLWriter.

Existen otras extensiones que puede que sean de tu interés, las cuales se encuentran categorizadas alfabéticamente y por categoría. También hay extensiones adicionales de PECL que podrían estar documentadas o no dentro del manual de PHP.

Ventajas y desventajas de PHP

Entre las **ventajas de PHP** se destaca su flexibilidad, el soporte amplio y la facilidad de aprendizaje. Puestos a hablar de las **desventajas de PHP**, analizaremos sus márgenes de seguridad.

Los siguientes son los principales **beneficios de PHP**:

- Es de **código abierto y de acceso gratuito**, por lo que no genera costos para los desarrolladores que trabajan con él.
- Es un lenguaje en **constante evolución**, con una gran comunidad que elabora cambios, elimina fallas, barre con los *bugs* y presenta optimizaciones periódicas.
- Su **comunidad es proactiva**, esto a la hora de compartir actualizaciones y ayuda entre colegas.

- Es capaz de **trabajar con grandes cantidades de datos**, aunque simplificándolos. Así, echa por tierra una de las grandes preocupaciones de los programadores: la complejidad de sus desarrollos se traduce en muchos datos. Con PHP la gestión se dinamiza y se mejora el rendimiento.
- Es **compatible con las principales bases de datos**, entre ellas MySQL, Interbase, Sybase, Oracle y SQLite.
- El **aprendizaje de PHP es intuitivo** y relativamente sencillo. Se lo considera entre los lenguajes de programación más fáciles de aprender. En este punto, entra en juego la mencionada comunidad que ofrece soporte, además de la disponibilidad de [cursos de programación especializados](#), tutoriales, manuales, etcétera.
- Finalmente, si bien PHP es simple para principiantes, también ofrece muchas **opciones para los programadores más avanzados**.

¿Tiene **desventajas PHP**? Aguzando la mirada, es posible encontrar algunas.

La facilidad de aprendizaje y uso de PHP puede derivar en malas prácticas: siendo que es relativamente sencillo usarlo, en ocasiones se emplea a este lenguaje sin los conocimientos necesarios para que los desarrollos sean robustos.

Por eso es relevante acudir a programadores profesionales, que puedan dar muestra de su trabajo y experiencia.

Sin embargo, huelga decir que aquella no es una desventaja de PHP *per se*, sino un mal uso de esa solución.

Por último, **¿qué tan seguro es PHP?**

De acuerdo al sitio *DesarrolloWeb.com*, [las primeras versiones del lenguaje, previas a PHP 3, tenían algunas configuraciones potencialmente inseguras](#), pero en las ediciones actuales no están presentes.

Por lo demás, PHP podría provocar inyecciones de SQL y derivar en ataques XSS. Si bien estos problemas pueden existir, el lenguaje ofrece mecanismos de protección y, así, una eventual vulnerabilidad no puede achacarse a PHP sino al desarrollador que no tomó los recaudos necesarios.

¿Qué se necesita para desarrollar en PHP?

Para **desarrollar en PHP** es preciso hacerlo desde computadoras con Linux, Windows o Mac, tal como lo mencionamos en el presente artículo

En la sección [¿En qué plataformas corre PHP?](#) También dijimos cuáles son los requisitos para programar con este lenguaje.

Lo cierto es que en el mercado actual son muy requeridos los programadores con experiencia en aplicaciones web y, en tal contexto, PHP asoma como uno de los principales recursos.

Siguiendo lo dicho anteriormente, la curva de [aprendizaje de PHP](#) es relativamente rápida, aunque siempre es recomendable la capacitación profesional para aprovechar al máximo los beneficios de la herramienta.

Por eso, desde Coderhouse, esperamos que puedas inscribirte en el curso de [programación backend](#) y aprendas todo lo necesario sobre el lenguaje de programación PHP. ¿Qué esperas para inscribirte?