

|  |  |
| --- | --- |
| fundamentos de PHP  11/octubre/2024 | BREVE DESCRIPCION  PHP es un lenguaje de programación interpretado​ del lado del servidor y de uso general que se adapta especialmente al desarrollo web.​  AUTOR: ISABELLA DAZA VERGARA  DOCENTE: FELIPE RODRIGUEZ  TPS2-123 |

Contenido

[1 Título Actividad 3.2 Fundamento de PHP 1](#_Toc180122724)

[2 Descripción de las actividades 1](#_Toc180122725)

[2.1 Que es convertir el código PHP a bytecode 1](#_Toc180122726)

[2.2 Ejemplo de código PHP a bytecode 2](#_Toc180122727)

[2.3 Descripción de como funciona el compilador JIT 2](#_Toc180122728)

[2.4 Servidor Web 3](#_Toc180122729)

[2.4.1 Que es un servidor web 3](#_Toc180122730)

[2.4.2 Descripción de la arquitectura del servidor web apache 3](#_Toc180122731)

[2.5 Descripción General entorno de desarrollo de visual estudio code. 4](#_Toc180122732)

[3.Extensiones de PHP 4](#_Toc180122733)

# Título Actividad 3.2 Fundamento de PHP

# Descripción de las actividades

¿Qué significa bytecode?

El bytecode es un tipo de código de programa que se compila y ejecuta en sistemas informáticos llamados VM (máquinas virtuales, por sus siglas en inglés). Los programadores pueden usar el bytecode en su forma original en cualquier plataforma en la que opere la VM, por lo que el código es independiente de la plataforma.

## Que es convertir el código PHP a bytecode

R// Convertir código PHP a bytecode implica transformar el código fuente de PHP en una representación más eficiente para la ejecución por parte del motor PHP, eliminando la necesidad de recompilar el código cada vez que se ejecuta.

**El bytecode es un formato** intermedio que es más cercano al código de máquina, pero aún independiente del hardware específico. En el caso de PHP, esto optimiza el rendimiento ya que el código ya ha sido analizado y transformado en instrucciones más directas para la máquina virtual de PHP (Zend Engine). Esto ahorra tiempo de procesamiento, ya que el código no tiene que ser reanalizado o interpretado completamente en cada solicitud.

**¿Cómo se realiza esto en PHP?** PHP por defecto compila el código en bytecode cuando se ejecuta, pero normalmente este bytecode no se almacena de manera persistente. Sin embargo, usando cachés de bytecode como OPcache, el bytecode compilado puede almacenarse en memoria y reutilizarse, mejorando notablemente la velocidad de ejecución de aplicaciones PHP.

**Un proceso típico es:**

El código PHP es compilado a bytecode cuando es ejecutado.

Opcionalmente, el bytecode puede ser almacenado usando extensiones como OPcache.

El bytecode es ejecutado por el Zend Engine (el motor interno de PHP).

En resumen, convertir a bytecode mejora el rendimiento de aplicaciones PHP al reducir la necesidad de recompilación del código, particularmente cuando se combina con sistemas de caché.

## Lenguaje PHP básicoEjemplo de código PHP a bytecode

R//

## Descripción de como funciona el compilador JIT

R// JIT, o la compilación JIT (just-in-time) es una forma de **ejecutar el código informático** que implica la compilación durante la ejecución de un programa, en tiempo de ejecución, y no antes de la ejecución. Normalmente se trata de **traducir el código fuente**, o **bytecode, a lenguaje de máquina**. ¿Será una respuesta de PHP a WebAssembly?

JIT es una estrategia compleja — pero no “horrible” — de complicación que toma una representación intermedia de código y la convierte en código de máquina dependiente de la arquitectura, en tiempo de ejecución.

Lo que hace JIT en PHP es hacer que Zend VM ya no sea el anfitrión de tu código, si no la CPU directamente — ¿qué te parece?

El compilador JIT compila opcode en código de máquina y lo ejecuta, envés de decirle a Zend VM que lo ejecute. — ¿Es el adios al interprete?

Desde PHP 7 que la comunidad PHP está centrada en el rendimiento — Sergio nos comentaba hace pocos meses sobre cómo mejorar el rendimiento con PHP 7.4-, sobre todo después del nacimiento del proyecto HHVM de Facebook. HHVM utiliza JIT y los resultados son los que todos conocemos.

No obstante, vale la pena referir que JIT puede no hacer de tu sitio web un site más rápido. ¿Por qué pasaría eso? Sencillo: Las aplicaciones desarrolladas en PHP hacen un uso intensivo de entrada/salida (p.e. lectura/escritura en base de datos), mientras que JIT depende del procesador.

JIT abre las puertas de PHP para que se pueda utilizar como un lenguaje de alto rendimiento — te imaginas usar JIT con Swoole ? — fuera de la web, abriendo así la puerta a ciencias de la moda como:

El Machine Learning, el renderizado 2D/3D, el análisis de datos

## Servidor Web

### Que es un servidor web

R// un servidor web es un ordenador que almacena, procesa y entrega archivos de sitios web a los usuarios desde un navegador. Los servidores web están formados por hardware y software que utilizan el **Protocolo de Transferencia de Hipertexto (HTTP)** para responder a las peticiones de los usuarios de la web realizadas a través de la World Wide Web.

A través de este proceso, los servidores web cargan y entregan la página solicitada al navegador del usuario, como Google Chrome.

**la mayoría de los servidores web ofrecen las siguientes características**

Registro de archivos.

Autenticación.

Limitación del ancho de banda.

Espacio de almacenamiento.

Lenguaje de programación.

**¿Para qué sirve un servidor web?**

1. Hospedar múltiples sitios o aplicaciones web.
2. Procesar solicitudes de Protocolo de Transferencia de Archivos (FTP).
3. Enviar y recibir correos electrónicos.

### Descripción de la arquitectura del servidor web apache

R// Apache es un servidor web de de código abierto, multiplataforma y gratuito.

Este web server es uno de los más utilizados en el mundo, actualmente el 43% de los sitios webs funcionan con él.

La función esencial del servidor Apache es servir las webs alojadas en el servidor a los diversos navegadores como Chrome, Firefox, Safari…

Apache consigue que la comunicación entre el servidor web y el cliente web (usuario que solicita la información) sea fluida y constante.

Haciendo que cuando un usuario haga una petición HTTP a través de navegador para entrar a una web o URL específica, Apache devuelva la información solicitada a través del protocolo HTTP.

## Descripción General entorno de desarrollo de visual estudio code.

R// Visual Studio es una herramienta de desarrollo eficaz que permite completar todo el ciclo de desarrollo en un solo lugar. Es un entorno de desarrollo integrado (IDE) completo que puede usar para escribir, editar, depurar y compilar el código.

es un editor de código fuente ligero pero eficaz que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux.

**Con Visual Studio, puede crear:**

Aplicaciones de . NET Core que se ejecutan en Windows, macOS y Linux.

Aplicaciones móviles para iOS, Android y Windows en C# y F# medianteXamarin.

Juegos 2D y 3D en C# mediante Visual Studio Tools para Unity.

Aplicaciones de C++ nativas para dispositivos iOS, Android y Windows.

# 3.Extensiones de PHP

**PHP IntelliSense**

PHP IntelliSense es una extensión es fundamental, porque te aporta mucha velocidad de escritura gracias al «intellisense». Mejora la experiencia de codificación al proporcionar sugerencias inteligentes de código y completar automáticamente fragmentos de código mientras escribes en PHP.

**PHP Intelephense**

La extensión PHP Intelephense ofrece características avanzadas para desarrolladores PHP, como autocompletado inteligente, análisis estático y soporte para PHPDoc. Ayuda a detectar errores y a escribir código más eficiente.

**PHP DocBlocker**

PHP DocBloker simplifica la creación de comentarios tipo DocBlock para sus funciones y métodos PHP. Con solo presionar un atajo de teclado, puede generar rápidamente documentación legible y útil para su código.

**Composer**

Composer es una herramienta esencial para administrar dependencias en proyectos PHP. Esta extensión facilita la ejecución de comandos de Composer directamente desde VSCode, lo que simplifica la gestión de paquetes y la instalación de dependencias.

**PHP Debug**

PHP Debug es una extensión esencial para la depuración de código PHP. Permite conectar VSCode a un servidor de depuración PHP y proporciona herramientas para inspeccionar variables, establecer puntos de interrupción y realizar un seguimiento detallado del flujo de ejecución.

**PHP Code Sniffer**

Asegura que tu código PHP cumpla con los estándares de codificación establecidos. Puedes personalizar las reglas según las necesidades de tu proyecto.

**PHP CS Fixer**

Automatiza la corrección de problemas de formato en tu código PHP, manteniéndolo limpio y coherente con facilidad.

**Laravel Blade Snippets:**

Resaltado de sintaxis: Facilita la lectura y escritura de plantillas Blade mediante el resaltado de sintaxis.

Fragmentos de código (snippets): Proporciona fragmentos de código predefinidos para estructuras comunes de Blade, como bucles y condiciones.

Autocompletado: Sugerencias de código específicas para Laravel Blade, mejorando la eficiencia y precisión al escribir plantillas.

**Laravel Extra Intellisense**

Otra extensión útil para desarrolladores de Laravel es Laravel Extra Intellisense, que mejora el autocompletado y el análisis de código específico para Laravel.

Autocompletado avanzado: Proporciona sugerencias de código basadas en las clases y métodos de Laravel, mejorando la eficiencia y precisión al escribir código.

Análisis de código: Detección de errores y sugerencias de mejora en tiempo real, ayudándote a mantener un código limpio y libre de errores.

Integración con Laravel: Optimizada específicamente para trabajar con el framework Laravel, proporcionando una experiencia de desarrollo fluida.

## 3.1 tabla

|  |  |
| --- | --- |
| **Extensión / Incorporación** | **Descripción / funciones y clases e interfaces** |
| **Extensiones básicas** |  |
| [Manejo de variables](http://php.net/manual/es/book.var.php) / núcleo | Funciones para el manejo de **variables**\: boolval, empty, floatval, get\_defined\_vars, gettype, intval, is\_array, is\_callable, is\_null, isset, print\_r, serialize, unset, unserialize, var\_dump... |
| [Manejo de funciones](http://php.net/manual/es/book.funchand.php) / núcleo | Funciones para el manejo de **funciones**\: call\_user\_func\_array, call\_user\_func, func\_get\_args, func\_num\_args, function\_exists, get\_defined\_functions... |
| [Strings](http://php.net/manual/es/book.strings.php) / núcleo | Funciones para la manipulación de **strings**\: chr, echo, explode, htmlentities, htmlspecialchars, implode, join, md5, nl2br, print, printf, str\_repeat, str\_replace, str\_shuffle, stripos, strlen, strtoupper, substr, ucfirst, trim... |
| [Arrays](http://php.net/manual/es/book.array.php) / núcleo | Funciones para interactuar con **arrays** y manipularlos\: array\_chunk, array\_diff, array\_keys, array\_map, array\_push, array\_rand, array\_replace, array\_search, array\_shift, array\_pop, array\_merge, array\_walk, count, each, extract, list, reset, sort... |
| [Información de clases y objetos](http://php.net/manual/es/book.classobj.php) / núcleo | Permiten obtener información sobre **clases** y **objetos**\: \_\_autoload, call\_user\_method, class\_alias, class\_exists, get\_called\_class, get\_class, get\_object\_vars, is\_a, get\_declared\_classes, method\_exists... |
| [Directorios](http://php.net/manual/es/book.dir.php) / núcleo | Funciones para trabajar con **directorios**. **Clases**: Directory. **Functions**\: chdir, chroot, closedir, dir, getcwd, opendir, scandir... |
| [Sistema de ficheros](http://php.net/manual/es/book.filesystem.php) / núcleo | Funciones para trabajar con **archivos**\: basename, chmod, chown, dirname, fclose, feof, file\_exists, file\_get\_contents, file\_put\_contents, file, filesize, fopen, fwrite, is\_dir, is\_readable, is\_uploaded\_file, mkdir, pathinfo, readfile, rmdir, tmpfile... |
| [Fecha/Hora](http://php.net/manual/es/book.datetime.php) / núcleo | Funciones para obtener **fecha y hora** que **dependen del servidor**. **Clases**: DateTime, DateTimeImmutable, DateTimeZone, DateInterval, DatePeriod. **Functions**\: checkdate, date\_add, date\_create, date\_format, date\_time\_set, date, localtime, microtime, strtotime, time... (muchas funciones son alias de las clases) |
| [Sesiones](http://php.net/manual/es/book.session.php) / núcleo | Funciones de **sesiones**, formas de preservar información para accesos subsiguientes. **Clases**: SessionHandler. **Interfaces**: SessionHandlerInterface. **Functions**\: session\_abort, session\_encode, session\_id, session\_name, session\_start, session\_status, session\_unset... |
| [Mail](http://php.net/manual/es/book.mail.php) / núcleo | Función que permite **enviar correos**\: mail |
| [PDO](http://php.net/manual/es/book.pdo.php) / bundled | La extensión **PDO** (PHP Data Objects) define una interfaz ligera para acceder a bases de datos en PHP. Proporciona una capa de abstracción de acceso a datos que emplea las mismas funciones y consultas independientemente de la base de datos que se utilice. **Clases**\: PDO, PDOStatement, PDOException. **Controladores**\: MS SQL Server, Mysql, Oracle, PostgreSQL, SQLite... |
| **Afectan al comportamiento de PHP** |  |
| [APC](http://php.net/manual/es/book.apc.php) / Pecl | Alternative PHP Cache, código de operación de **caché libre** y abierto para PHP. Clases: APCIterator. Functions: apc\_add, aps\_bin\_dump, apc\_cache\_info, apc\_clear\_cache, apc\_exists, apc\_store... |
| [Manejo de errores](http://php.net/manual/es/book.errorfunc.php) / núcleo | Funciones para el manejo y registro de **errores**\: debug\_backtrace, error\_clear\_last, error\_get\_last, error\_log, set\_error\_handler, trigger\_error... |
| [OPCache](http://php.net/manual/es/book.opcache.php) / bundled | **Mejora el rendimiento de PHP** almacenando el código de bytes en un script precompilado en la memoria compartida, eliminando la necesidad de que PHP cargue y analice los script en cada petición. Functions\: opcache\_get\_configuration, opcache\_get\_status, opcache\_is\_script\_cached... |
| [Control de la salida](http://php.net/manual/es/book.outcontrol.php) / núcleo | Funciones que permiten controlar **cuándo se envía la salida**. Functions\: flush, ob\_clean, ob\_flush, ob\_get\_contents, ob\_get\_status, ob\_start... |
| [Opciones e información de PHP](http://php.net/manual/es/book.info.php) / núcleo | Funciones que proporcionan **información sobre PHP**. Functions\: extension\_loaded, gc\_enable, get\_defined\_constants, get\_extension\_funcs, get\_include\_path, get\_loaded\_extensions, getenv, memory\_get\_usage, phpinfo, phpversion, set\_time\_limit, version\_compare... |
| **Extensiones de compresión y archivos** |  |
| [Bzip2](http://php.net/manual/es/book.bzip2.php) / externa | Módulo para leer y escribir archivos comprimidos con **bzip2** (.bz2). Functions\: bzclose, bzcompress, bzdecompress, bzerror, bzflush, bzopen... |
| [Zlib](http://php.net/manual/es/book.zlib.php) / bundled | Módulo para leer y escribir ficheros comprimidos con **gzip** (.gz). Functions\: gzclose, gzcompress, gzencode, gzfile, gzopen, gzread, gztell, gzwrite... |
| [LZF](http://php.net/manual/es/book.lzf.php) / pecl | **LZF** es un algoritmo de compresión muy rápido, ideal para ahorrar espacio con un ligero costo en velocidad. Utiliza la [**librería liblzf**](http://oldhome.schmorp.de/marc/liblzf.html). Functions\: lzf\_compress, lzf\_decompress, lzf\_optimized\_for |
| [Zip](http://php.net/manual/es/book.zip.php) / externa | Extensión que permite leer o escribir **archivos Zip** y los archivos de dentro. Clases: ZipArchive. Functions\: zip\_close, zip\_entry\_filesize, zip\_entry\_name, zip\_entry\_name, zip\_entry\_read, zip\_open, zip\_read... |
| **Extensiones criptográficas** |  |
| [Hash](http://php.net/manual/es/book.hash.php) / núcleo | Motor para cifrar mensajes (algoritmos hash). Functions\: hash\_algos, hash\_copy, hash\_file, hash\_init, hash\_update, hash... |
| [Mcrypt](http://php.net/manual/es/book.mcrypt.php) / externas | Interfaz para la biblioteca mcrypt, admite una gran variedad de algoritmos de bloques como DES, TRIpleDES, Blowfish, 3-WAY, SAFER-SK64, SAFER-SK128, TWOFISH, TEA, RC2 y GOST en los modos de cifrado CBC, OFB, CBF y ECB. Functions\: mcrypt\_cbc, mcrypt\_cfb, mcrypt\_decrypt, mcrypt\_encrypt, mcrypt\_generic, mcrypt\_list\_algorithms, mcrypt\_list\_modes, mcrypt\_module\_open, mcrypt\_ofb... |
| [OpenSSL](http://php.net/manual/es/book.openssl.php) / externas | Módulo que emplea las funciones de [OpenSSL](http://www.openssl.org/) (sólo algunas) para la generación y verificación de firmas y para encriptar y desencriptar datos. Functions:\ openssl\_csr\_export, openssl\_csr\_new, openssl\_csr\_sign, openssl\_decrypt, openssl\_digest, openssl\_encrypt, openssl\_open, openssl\_pkey\_free, openssl\_seal, openssl\_sign, spenssl\_spki\_new, openssl\_verify... |
| [Hash de contraseñas](http://php.net/manual/es/book.password.php) / núcleo | Proporciona una envoltura fácil de usar sobre [***crypt()***](http://php.net/manual/es/function.crypt.php) para hacerla sencilla para crear y administrar contraseñas de forma segura. Functions\: password\_get\_info, password\_hash, password\_needs\_rehash, password\_verify |
| **Soporte para el lenguaje humano y codificación de caracteres** |  |
| [iconv](http://php.net/manual/es/book.iconv.php) / bundled | Este módulo tiene una interfaz para la conversión de **conjuntos de caracteres iconv**. Se puede convertir un conjunto de caracteres local por otro, el cual puede ser **Unicode**. Es recomendable instalar la librería [GNU libiconv](http://www.gnu.org/software/libiconv/). Functions\: iconv\_get\_encoding, iconv\_mime\_encode, iconv\_strlen, iconv\_strpos, iconv\_substr, iconv... |
| [intl](http://php.net/manual/es/book.intl.php) / bundled | Extensión para la internacionalización ***intl*** es una envoltura para la librería [ICU](http://site.icu-project.org/), permitiendo realizar un cotejo que cumple con el [UCA](http://www.unicode.org/reports/tr10/) y formatear fecha/hora/número/moneda en los scripts. **Clases**: Collator, NumberFormatter, Locale, Normalizer, MessageFormatter, IntlCalendar, IntlTimeZone, IntlDateFormatter, ResourceBundle, Spoofchecker, Transliterator, IntlChar, IntlException... |
| [mbstring](http://php.net/manual/es/book.mbstring.php) / bundled | Los **esquemas de codificación multibyte** se desarrollaron para expresar más de 256 caracteres en el sistema de codificación regular a nivel de bits. Cuando se manipulan strings en una codificación multibyte, es necesario utilizar funciones especiales. ***mbstring*** proporciona funciones específicas para cadenas de texto multibyte que ayudan a tratar codificaciones multibyte en PHP. Functions\: mb\_convert\_case, mb\_detect\_encoding, mb\_encoding\_aliases, mb\_ereg\_replace, mb\_ereg\_search\_init, mb\_ereg, mb\_parse\_str, mb\_send\_mail, mb\_split, mb\_strlen, mb\_strrpos, mb\_substr... |
| **Procesamiento y generación de imágenes** |  |
| [Exif](http://php.net/manual/es/book.exif.php) / bundled | Con la extensión Exif, Exchangeable image information, se puede trabajar con metadatos de imágenes (ej: funciones para leer metadatos de fotografías tomadas con cámaras digitales con la información almacenada en las cabeceras de imágenes JPEG y TIFF). Functions\: exif\_imagetype, exif\_read\_data, exif\_tagname, exif\_thumbnail... |
| [GD](http://php.net/manual/es/book.image.php) / bundled | PHP también se puede usar para crear y manipular ficheros de imágenes en diferentes formatos: GIF, PNG, JPEG, WBMP, XPM. PHP puede transferir flujos de imagen directamente al navegador. Functions\: gd\_info, getimagesize, imageaffine, imagearc, imagecolorat, imagecolormatch, imagecolorset, imagecopy, imagecreate, imagecreatefromgd, imagecrop, imagedestroy, imagefill, imageflip, imagegif, imagejpeg, imageline, imagescale... |
| [ImageMagick](http://php.net/manual/es/book.imagick.php) / pecl | Extensión nativa de PHP para crear y modificar imágenes con la API **ImageMagick** (software para crear, editar y componer imágenes de mapa de bits). Puede leer, convertir y escribir imágenes en más de 100 formatos. **Clases**: Imagick, ImagickDraw, ImagicPixel, ImagickPixelIterator, ImagickKernel... |
| **Servicios web** |  |
| [OAuth](http://php.net/manual/es/book.oauth.php) / pecl | Extensión que provee un cliente y un servicio **OAuth** (protocolo de autorización basado en **HTTP** que permite a las aplicaciones garantizar el acceso a datos sin tener alojado un nombre de usuario y contraseña). **Clases**: OAuth, OAuthProvider |
| [SOAP](http://php.net/manual/es/book.soap.php) / externas | La extensión **SOAP** se utiliza para escribir servidores y clientes SOAP. **Clases**: SOAPClient, SoapServer, SoapFault, SoapHeader, SoapParam, SoapVar... |
| **Manipulación XML** |  |
| [DOM](http://php.net/manual/es/book.dom.php) / externas | La extensión **DOM**, Document Object Model, permite manipular documentos XML mediante la API DOM. **Clases**: DOMAttr, DOMCdataSection, DOMCharacterData, DOMComment, DOMDocument, DOMElement, DOMEntity, DOMNode... |
| [SimpleXML](http://php.net/manual/es/book.simplexml.php) / externas | Proporciona un conjunto de herramientas para **convertir XML a un objeto** que pueda ser procesado con selectores de propiedades e iteradores de arrays. **Clases**: SimpleXMLElement, SimpleXMLIterator. Functions\: simplexml\_import\_dom, simplexml\_load\_file, simplexml\_load\_string... |
| [Analizador XML](http://php.net/manual/es/book.xml.php) / externas | Extensión que implementa el soporte para expat con herramientas para **analizar** (no validar) **documentos XML**. Functions\: utf8\_encode, xml\_error\_string, xml\_get\_current\_line\_number, xml\_get\_error\_code, xml\_parse, xml\_set\_default\_handler, xml\_set\_object |
| [XMLReader](http://php.net/manual/es/book.xmlreader.php) / externas | **XMLReader** es un **analizador de XML**. El lector actúa como un cursor yendo hacia delante en el flujo del documento y deteniéndose en cada nodo. **Clases**: XMLReader |
| [XMLWriter](http://php.net/manual/es/book.xmlwriter.php) / externas | Envuelve la API XMLWriter de libxml. Representa un escritor que provee medios no almacenados en caché de sólo avance para la generación de flujos o ficheros con datos XML. **Clases**: XMLWriter |
| **Otras extensiones básicas** |  |
| [Curl](http://php.net/manual/es/book.curl.php) / externas | Librería ***libcurl*** que permite **conectarse y comunicarse con diferentes tipos de servidores y protocolos**: http, https, ftp, gopher, telnet, dict, file y ldap. También admite certificados HTTPS, HTTP, POST, HTTP PUT, subidas mediante FTP (también es posible hacerlo con la extensión FTP de PHP), subidas con formularios HTTP, proxies, cookies y autentificación. **Clases**: CURLFile. Functions\: curl\_close, curl\_errno, curl\_error, curl\_exec, curl\_init, curl\_multi\_exec, curl\_multi\_init, curl\_pause, curl\_reset, curl\_setopt, curl\_strerror... |
| [FTP](http://php.net/manual/es/book.ftp.php) / bundled | Las funciones de esta extensión implementan el acceso por parte del cliente a los servidores mediante **FTP**, el [Protocolo de Transferencia de Ficheros](http://www.faqs.org/rfcs/rfc959.html). Functions\: ftp\_alloc, ftp\_chdir, ftp\_chmod, ftp\_close, ftp\_connect, ftp\_exec, ftp\_get, ftp\_login, ftp\_mkdir, ftp\_put, ftp\_quit, ftp\_raw, ftp\_rename, ftp\_rmdir... |
| [LDAP](http://php.net/manual/es/book.ldap.php) / externas | El **LDAP**, Lightweight Directory Acess Protocol, se utiliza para acceder a servidores de directorio. Un **directorio es un tipo especial de base de datos** que mantiene información en una estructura de árbol. Functions\: ldap\_add, ldap, bind, ldap\_close, ldap\_compare, ldap\_connect, ldap\_delete, ldap\_error, ldap\_free\_result, ldap\_get\_values, ldap\_list, ldap\_read... |
| [GeoIP](http://php.net/manual/es/book.geoip.php) / pecl | La extensión GeoIP permite buscar la **localicación de una dirección IP** (ciudad, país, estado, longitud, latitud, ISP, tipo de conexión...). Functions\: geoip\_continent\_code\_by\_name, geoip\_country\_code\_by\_name, geoip\_database\_info, geoip\_id\_by\_name, geoip\_netspeedcell\_by\_name, geoip\_region\_by\_name, geoip\_time\_zone\_by\_country\_and\_region... |
| [JSON](http://php.net/manual/es/book.json.php) / bundled | Extensión que implementa el formato de intercambio de datos **JSON** [Javascript Object Notation JSON](http://www.json.org/). **Interface**: JsonSerializable. Functions:\ json\_decode, json\_encode, json\_last\_error\_msg, json\_last\_error... |
| [Reflection](http://php.net/manual/es/book.reflection.php) / núcleo | **API de reflexión** con capacidad para realizar ingeniería inversa de clases, interfaces, funciones, métodos y extensiones. Ofrece también formas de obtener comentarios de documentación de funciones, clases y métodos. **Clases**: Reflection, ReflectionClass, ReflectionExtension, ReflectionFunction, ReflectionMethod, ReflectionProperty... |
| [SPL](http://php.net/manual/es/book.spl.php) / núcleo | **Biblioteca Estándar de PHP**, Standard PHP Library, colección de interfaces y clases pensadas para solucionar problemas comunes. Estructuras de datos, Iteradores, Interfaces, Excepciones, Manejo de ficheros. Functions\: class\_implements, class\_uses, iterator\_count, spl\_autoload\_register, spl\_classes... |
| [Tokenizer](http://php.net/manual/es/book.tokenizer.php) / núcleo | Funciones que proporcionan una interfaz para la **creación de tokens de PHP**. Con esta extensión, integrada en el Motor Zend, se pueden escribir herramientas propias de análisis o de modificación de código fuente de PHP sin tener que hacer frente a las especificaciones del lenguaje a nivel léxico. Functions\: token\_get\_all, token\_name |
| [URLs](http://php.net/manual/es/book.url.php) / núcleo | Funciones para tratar con **URLs**: codificación, decodificación y conversión. Functions\: get\_headers, get\_meta\_tags, parse\_url, urldecode, urlencode... |
| [Memcached](http://php.net/manual/es/book.memcached.php) / pecl | Sistema de alto rendimiento para el almacenamiento de objetos en caché de memoria distribuida, pensado para acelerar aplicaciones web dinámicas aliviando la carga de bases de datos. **Clases**: Memcached, MemcachedException |
| [SSH2](http://php.net/manual/es/book.ssh2.php) / pecl | Enlaza a la biblioteca [libssh2](http://libssh2.org/) que provee **acceso a recursos sobre una máquina remota** utilizando una **vía de transporte criptográfica segura**. Functions\: ssh2\_auth\_agent, ssh2\_auth\_none, ssh2\_connect, ssh2\_exec, ssh2\_fingerprint, ssh2\_scp\_recv, ssh2\_sftp\_mkdir, ssh2\_sftp\_rename, ssh2\_sftp... |