

Estás en: [Artesanía de software \(https://jrgonzalez.es/blog\)](https://jrgonzalez.es/blog) > [PHP \(https://jrgonzalez.es/category/php\)](https://jrgonzalez.es/category/php) > **error\_log - guardar log en PHP**

# Error Log

Creado por [Juan Ramón González Morales \(https://jrgonzalez.es/author/pvto-amo\)](https://jrgonzalez.es/author/pvto-amo) && alojado en [PHP \(https://jrgonzalez.es/category/php\)](https://jrgonzalez.es/category/php).

En esta entrada voy a hablar de «**error\_log**» la opción para almacenar información tales como variables para depurar, guardar errores, comprobar resultados o queries ... en **PHP**. Usar el log para debuggar errores en php con **error\_log** es lo más natural del mundo.

## ¿Qué es un log?

Antes de nada, vamos a ver qué es un log. Un fichero de log es un archivo, normalmente de texto, donde se guardan datos y estadísticas de la traza de trabajo de una aplicación. Esta información no está visible para los usuarios de la misma, pero sí para los programadores que trabajan con ella, tanto para analizar datos como para detectar errores.

Los datos que normalmente se pueden guardar pueden ser:

Estado del programa

Accesos al mismo

Estado de las redes

Seguridad

Información que queremos analizar

...

Eso son algunos de los usos que se le pueden dar. Los logs dan para un post en sí mismos, pero creo que era necesario dar una pequeña explicación de qué es un log antes de continuar.

Normalmente, durante el desarrollo en PHP, uno siempre necesita debuggear y una de las cosas más usadas es dejar trazas en el log de PHP. Esto es ir almacenando información en un log para comprobar el estado del sistema, o también para dejar rastro de por dónde va pasando la aplicación en desarrollo. Para ello usamos la función de PHP **error\_log** que está pensada para eso mismo.

## Uso básico de la función error\_log de PHP

Ahora sí, vamos a ver **cómo trabajar con error\_log en PHP**.

Para ello, podemos usar la función **de php error\_log** si queremos escribir algo en el fichero de log de PHP.

Por defecto se añade en el fichero que esté configurado en php.ini para almacenar la información y los errores de PHP. Aunque, podemos añadir un parámetro a la función y escribir en el nuestros logs en diferentes ficheros de logs de PHP que nosotros deseemos.

Para almacenar algo en el log usando **PHP**, se hace de una manera muy simple, usando la función de PHP **error\_log**.

```
// Variable a salvar
$var = "HOLA";
// Dirección del fichero de log que queremos
$path = "log.log";
// Salvar en el log
error_log($var, 3, $path);
```

A continuación vamos a explicar el código anterior para almacenar en el log de PHP.

‘**\$var**’ es una variable con lo que queremos salvar en el log usando la función de **PHP error\_log**, entraré más en profundidad más adelante, para salvar objetos complejos y algunas situaciones excepcionales.

**\$path** es la ruta o path del fichero donde vamos a almacenar **\$var**.

Finalmente, llamando a «**error\_log(\$var, 3, \$path);**» se nos salvará **\$var** en nuestro fichero de logs.

NOTA:

Usamos **'3'** por defecto para indicarle a la función de «**error\_log**» que el destino es un fichero de texto.

Puedes encontrar más información de error\_log en la [página oficial de error\\_log en PHP.net](https://php.net/manual/es/function.error-log.php) (<https://php.net/manual/es/function.error-log.php>).

## Activar error logging en php.ini

Hay que buscar esta directiva **log\_errors** y colocarla a **On**, por defecto suele estar desactivado .

```
; Besides displaying errors, PHP can also log errors to locations such as a
; server-specific log, STDERR, or a location specified by the error_log
; directive found below. While errors should not be displayed on productions
; servers they should still be monitored and logging is a great way to do that.
; Default Value: Off
; Development Value: On
; Production Value: On
; http://php.net/log-errors
log_errors = On
```

Una vez hecho esto, se empezará a almacenar información durante la ejecución de PHP.

## Combinando error\_log y print\_r

Pero por defecto no se guardará la mayoría de información de una manera legible al usuario, por lo que necesitamos adaptar un poco el uso que hacemos de **error\_log**.

**Usando print\_r para obtener un string legible para usuarios de nuestros objetos complejos.**

Esto es muy útil para arrays y stdClass objects principalmente, aunque aplicable a cualquier tipo de objetos complejos.

Pasando el parámetro **return a true** a la función print\_r nos devolverá un string legible de \$variable se guardará en la variable **\$variable\_as\_string**. Ejemplo:

```
$variable_as_string = print_r($variable, $return);
```

Con esta función, tendremos como resultado en `$variable_as_string` la información de la variable pasada a la función `print_r $variable` lista para ser guardada en nuestro log de PHP con la función `error_log`.

Más información de «***print\_r***» en la [página oficial de PHP sobre la función print\\_r](https://php.net/manual/es/function.print-r.php) (<https://php.net/manual/es/function.print-r.php>).

### Guardar array en error\_log

La manera más sencilla de guardar un array en el log de PHP es usando la salida función **`print_r`** como string y añadirlo al log. Tal cuál se ha explicado anteriormente, guardar un array como string en el log es tan sencillo como lo siguiente.

```
// Convertir nuestro array a un string
$array_as_string = print_r($array, true);
//Guardar nuestro string del array en el fichero log.log
error_log(print_r($test_object, true), 3, "log.log");
```

### Guardar objetos (stdClass) en error\_log

Para ello, vamos a hacer uso de «***print\_r***» al igual que hemos hecho antes para objetos. Así podemos guardar nuestro objeto de forma fácil de leer y de entender para el usuario.

Esto sería tan sencillo como usando el código siguiente en **PHP para guardar un objeto en el log**.

```
// Código de ejemplo para almacenar un objeto en el log
$test_object = new stdClass();
$test_object->var = "HOLA ";
$test_object->nombre = " PEDRO";
error_log(print_r($test_object, true), 3, "test_log.log");
```

Usando el código de arriba, tendremos en nuestro fichero de log un resultado como en la imagen siguiente.

En caso de no usar «***print\_r***» no se guardaría nada legible en el fichero de log para un objeto `stdClass` o una clase personalizada.

# Creando una función personalizada de error\_log

Durante el desarrollo es normal usar muchas veces las llamadas a funciones de log.

Una manera de encapsular y simplificar el código es realizar una función para ello, lo cual será más fácil de recordar y de usar, también evitaremos repetir código en distintos sitios y tendremos un código usable, reutilizable, más limpio y elegante.

Un ejemplo podría ser el siguiente.

```
// Función para almacenar una variable en el fichero de log.
public function save_in_log($data, $path = "log.log")
{
    error_log(print_r($data, true), 3, $path);
    error_log("\xA", 3, $path);
}

// Ejemplos de uso.

// Guardar un string en el fichero de log por defecto de la función.
save_in_log("HOLA");

// Guardar una variable en el fichero de log por defecto de la función.
save_in_log($var);

// Guardar un string en el fichero "test_log.log".
save_in_log("HOLA", "test_log.log");

// Guardar una variable en el fichero "test_log.log".
save_in_log($var, "test_log.log");
```

Como se puede ver, he personalizado una función y he añadido algunos ejemplos de uso.

Por defecto, después de guardar los datos he añadido un salto de línea. Así la siguiente vez que se ejecute la función aparecerá en una nueva línea.

## Otros datos interesantes a añadir al log

Podemos añadir la fecha y el tiempo que ha sido añadido.

Para ello podemos usar gmdate:

```
function addToLog($message)
{
    $logpath = "log.log";
    $logMessage = gmdate("Y-m-d\TH:i:s\Z") . " - LOG TESTING - " . $message .
    PHP_EOL;
    error_log($logMessage, 3, $logpath);
}
```

La función anterior nos dará un output como el siguiente:

```
2020-08-24T10:29:59Z - LOG TESTING - GET
2020-08-24T10:29:59Z - LOG TESTING - Array
(
)

2020-08-24T10:29:59Z - LOG TESTING - POST
2020-08-24T10:29:59Z - LOG TESTING - Array
(
    [mc_gross] => 10.00
    [protection_eligibility] => Eligible
    [tax] => 0.00
    [payment_date] => 03:29:53 Aug 24, 2020 PDT
    [payment_status] => Completed
)
```

## Enlaces de interés

<http://php.net/manual/es/function.error-log.php> (<https://php.net/manual/es/function.error-log.php>)

<http://php.net/manual/es/function.print-r.php> (<https://php.net/manual/es/function.print-r.php>)

## Otras entradas similares

[Facade Storage y FTP en Laravel](https://jrgonzalez.es/facade-storage-y-ftp-en-laravel) (<https://jrgonzalez.es/facade-storage-y-ftp-en-laravel>)

[Reemplazar bucles básicos por las funciones nativas de arrays en PHP](#)

[. \(https://jrgonzalez.es/reemplazar-bucles-basicos-por-las-funciones-nativas-de-arrays-en-php\)](https://jrgonzalez.es/reemplazar-bucles-basicos-por-las-funciones-nativas-de-arrays-en-php)

[Fechas como string en PHP](#)  [\(https://jrgonzalez.es/fechas-como-string-php\)](https://jrgonzalez.es/fechas-como-string-php)

[¿Cómo recortar y redimensionar imágenes en PHP?](#)  [\(https://jrgonzalez.es/recortar-redimensionar-imagenes-en-php-imagecopyresampled\)](https://jrgonzalez.es/recortar-redimensionar-imagenes-en-php-imagecopyresampled)

[String contain en PHP](#)  [\(https://jrgonzalez.es/string-contain-php\)](https://jrgonzalez.es/string-contain-php)

[¿Cómo recargar una página con JavaScript?](#)  [\(https://jrgonzalez.es/como-recargar-la-pagina-con-javascript\)](https://jrgonzalez.es/como-recargar-la-pagina-con-javascript)

[Resumen 2020](#)  [\(https://jrgonzalez.es/resumen-profesional-del-ano-2020\)](https://jrgonzalez.es/resumen-profesional-del-ano-2020)

## 2 Responses to “Error Log”

### Enlaces de interés

---

[Dragon Ball Mania - Todo sobre Dragon Ball en la red](#)  [\(https://jrgonzalez.es/dbmania/\)](https://jrgonzalez.es/dbmania/)

[Keyword Research tool by jrgonzalez.es](#)  [\(https://jrgonzalez.es/apps/kwr/\)](https://jrgonzalez.es/apps/kwr/)

[SEO URL scraper](#)  [\(https://jrgonzalez.es/apps/scraper/url/\)](https://jrgonzalez.es/apps/scraper/url/)

[Herramienta Copia - Pega online](#)  [\(https://clip.jrgonzalez.es\)](https://clip.jrgonzalez.es)

[CV Juan Ramón González Morales](#)  [\(https://jrgonzalez.es/files/resume.pdf\)](https://jrgonzalez.es/files/resume.pdf)

[JR Pronósticos - Los pronósticos de La Liga 2021 2022](#)  [\(https://quiniela.jrgonzalez.es\)](https://quiniela.jrgonzalez.es)

[Generador quiniela aleatoria](#)  [\(https://jrgonzalez.es/quinielaAleatoria.html\)](https://jrgonzalez.es/quinielaAleatoria.html)

---

© 2023 [jrgonzalez.es](https://jrgonzalez.es).  [\(https://jrgonzalez.es\)](https://jrgonzalez.es)    [Política de privacidad](https://jrgonzalez.es/aviso-legal)  [\(https://jrgonzalez.es/aviso-legal\)](https://jrgonzalez.es/aviso-legal)

Juan Ramón González Morales - Software Analyst