



# LA SEGURIDAD IMPORTA

21 trucos para tener tu  
WordPress seguro



# INTRODUCCIÓN

## INTRO

¿WordPress es seguro? Sí. Tiene actualizaciones regulares, posee una gran comunidad de desarrolladores detrás e incorpora las últimas novedades tecnológicas en materia de seguridad.

### **Entonces, ¿por qué una guía de seguridad?**

Porque la seguridad se basa en añadir dificultades extra a los atacantes, prevenir el fallo, rastrear cambios, prohibir accesos maliciosos y ocultar información sensible de nuestro sitio, entre otras tareas. WordPress también es una plataforma en continuo desarrollo, abierta para el desarrollo y para integraciones con terceros, utilizada por el 30% de Internet, lo que la hace objetivo de ciberdelincuentes. En resumen, porque queremos hacer nuestro sitio lo más seguro posible.

Muchos usuarios piensan que nunca van a ser atacados por la tipología o tamaño de su proyecto web, pero a veces el objetivo no es la propia web o sus contenidos, sino hacerse o utilizar los recursos de servidor de nuestro proyecto para sus fines, y los proveedores de hosting somos conscientes de ello y del desconocimiento que muchas veces existe en materia de seguridad.

En SiteGround además nos preocupa especialmente la seguridad, tanto la nuestra como la de nuestros clientes, por lo que, entre otras acciones, realizamos actualizaciones automáticas de sus instalaciones de WordPress, disponemos de sistemas de escaneo y filtrado preventivo, ponemos capas adicionales de seguridad activa y actualizamos nuestro software con los últimos parches de seguridad disponibles, de una manera rápida y transparente para nuestros clientes.

Utiliza esta guía para enriquecer tus conocimientos sobre seguridad y WordPress, aplica aquellos que pueden beneficiarte en tu proyecto, comparte, ayuda y comenta con los de alrededor.



1

**PROTEGE  
TUS ARCHIVOS Y  
LA BASE DE DATOS**

# 1. PREVIO A LA INSTALACIÓN



Antes de realizar cualquier instalación nueva de un sitio con WordPress, por supuesto piensa siempre en empezar tu proyecto con la última versión estable, conviene realizar dos sencillos pasos relacionados con la seguridad de tu sitio en el archivo `wp-config.php`: cambiar el prefijo de la base de datos y utilizar las claves de autenticación

Todas las instalaciones por defecto en WordPress contienen el prefijo `wp_` para la bases de datos, siempre la misma para todas, por lo que conviene cambiarlo para cada sitio web con el fin de prevenir de posibles ataques relacionados con nuestra base de datos.

Para cambiar el prefijo de las tablas de WordPress, simplemente debes cambiar en el archivo de configuración `wp-config.php` la siguiente línea por el prefijo que prefieras:

```
$table_prefix = 'wp_';
```

Por ejemplo:

```
$table_prefix = 'nuevositio_wp_';
```

Este cambio además te permitirá tener varias instalaciones de WordPress sobre la misma base de datos, siempre que no repitas el prefijo.

Si tu sitio ya está instalado, y no cambiaste el prefijo por defecto durante el proceso de instalación, tienes plugins disponibles como `Change Table Prefix` que te permiten realizar ese cambio de una manera fácil. También puedes realizarlo de manera manual, aunque no es muy recomendable si no estás familiarizado con realizar cambios en la base de datos.

WordPress posee unas claves secretas (denominadas `Keys` y `Salt`) que están almacenadas en el archivo `wp-config.php`, y protegen las sesiones abiertas encriptando los datos de sesión en la cookie de tu navegador, por lo que antes de comenzar con la instalación, genera esas claves secretas.

De la misma manera que con el prefijo, puedes cambiar las claves secretas sobre un sitio ya instalado, en cualquier momento, truco que te recomiendo llevar adelante cada cierto tiempo con el fin de invalidar las sesiones activas y forzar a todos los usuarios a realizar de nuevo el inicio de sesión.

Aunque puedes generar tus propias claves manualmente, te recomiendo utilizar el servicio oficial de WordPress en la siguiente URL y sustituir el contenido de esas claves por las de tu fichero wp-config.php:

<https://api.wordpress.org/secret-key/1.1/salt/>

Te dejo por último un pequeño truco sobre estas claves para aquellas instalaciones en producción donde quieras prohibir cualquier tipo de acceso al panel de administración, incluso conociendo usuario y contraseña, ya que se invalidan a cada microsegundo. Simplemente debes reemplazar vuestras claves en el fichero wp-config.php por estas:

```
define('AUTH_KEY',          microtime());
define('SECURE_AUTH_KEY',   microtime());
define('LOGGED_IN_KEY',     microtime());
define('NONCE_KEY',         microtime());
define('AUTH_SALT',         microtime());
define('SECURE_AUTH_SALT',  microtime());
define('LOGGED_IN_SALT',    microtime());
define('NONCE_SALT',        microtime());
```

Recuerda actualizar estas claves cada cierto tiempo, de manera preventiva o para eliminar sesiones abiertas.



## 2. TRAS LA INSTALACIÓN



Una vez terminada la instalación de tu nuevo sitio con WordPress, conviene eliminar el usuario que has utilizado para el proceso de instalación, usuario con perfil de administrador, crear un nuevo usuario con perfil de administrador, así como todos los usuarios necesarios.

Recuerda no utilizar nombres de usuario como admin o administrador, muy comunes en todas las instalaciones, así como no utilizar contraseñas débiles.

Deshabilita las notificaciones de pingbacks y trackbacks en tu panel de administración (Ajustes > Comentarios), ya que puede ser una entrada para posibles ataques masivos de denegación de servicio (DDoS) contra tu sitio.

Asimismo debes proteger ciertos archivos de ataques o intrusiones una vez completada la instalación, añadiendo las siguientes líneas en el archivo .htaccess, preferiblemente al inicio del archivo situado en el raíz de tu sitio:

```
#No listar directorios
```

```
Options - Indexes
```

```
#Bloquear archivos sensibles
```

```
<files .htaccess>
```

```
Order allow,deny
```

```
Deny from all
```

```
</files>
```

```
<files wp-config.php>
```

```
Order allow,deny
```

```
Deny from all
```

```
</files>
```

Una vez completada la instalación, existen más ficheros que no son necesarios y que conviene bloquear su acceso, por lo que debes crear un archivo nuevo .htaccess dentro del directorio /wp-admin y añadir estas líneas en él:

```
#Bloquear archivos instalación
```

```
<files install.php>
```

```
Order allow,deny
```

```
Deny from all
```

```
</files>
```

```
<files setup-config.php>
```

```
Order allow,deny
```

```
Deny from all
```

```
</files>
```

Tampoco viene mal repasar nuestro archivo robots.txt, situado en el directorio raíz de nuestro sitio, no sólo por su utilidad para indicar a los robots de búsqueda qué deben y qué no deben analizar de tu sitio, sino para que no aparezca información relacionada con nuestra instalación WordPress, por ejemplo, nuestra carpeta wp-admin.

### 3. PERMISOS DE FICHEROS Y CARPETAS



Para evitar que los atacantes puedan tomar control de tu instalación hay que asegurarse de que los archivos y carpetas que tenemos en WordPress tienen los permisos de acceso adecuados.

Habitualmente es posible cambiar los permisos mediante un cliente FTP o con alguna opción que nuestro proveedor de alojamiento incluya en los paneles de administración. En el caso de SiteGround, tienes disponible una opción para poder cambiar los permisos de los ficheros y carpetas de una manera sencilla.

La encontrarás en tu panel de control: “Herramientas de WordPress”, “Kit de herramientas de WordPress”, “Arreglar permisos”.

- *Los permisos para todas las carpetas deben ponerse a 755*
- *Los permisos para todos los archivos deben ponerse a 644*

Para restringir todavía más, existen dos archivos especiales en nuestra configuración de WordPress que podemos asegurar de la siguiente manera:

- *Archivo `wp-config.php`: poner permisos de archivo en 600*
- *Archivo `.htaccess`: poner permisos de archivo en 604*

Estos números que te he puesto hacen referencia a la codificación de los permisos de lectura, escritura y ejecución definidos en sistemas operativos Unix.

## 4. BLOQUEAR PHP EN CARPETAS



Aunque en las instalaciones de WordPress, por defecto, no se permiten subir archivos PHP mediante el panel de administración, conviene bloquear la posibilidad de que se ejecute código PHP en esa carpeta, y de paso, asegurar también la ejecución innecesaria de código PHP en otras carpetas utilizadas por WordPress y que no deben accederse directamente.

Para ello, debemos crear un nuevo archivo `.htaccess` dentro de nuestros directorios `/wp-content/uploads`, `/wp-content/plugins` y `/wp-content/themes` y añadir las siguientes líneas a dicho archivo con el fin de bloquear las ejecuciones PHP:

```
<Files *.php>
```

```
deny from all
```

```
</Files>
```

**Nota:** recuerda que después de cada modificación en un archivo `.htaccess`, debes comprobar el correcto funcionamiento de tu instalación, borrando previamente la caché de tu sitio para confirmar que las reglas añadidas funcionan de manera adecuada.



## 5. DESHABILITAR LA EDICIÓN DE FICHEROS EN WORDPRESS



En este truco nos centraremos en poner una capa de seguridad dentro del propio panel de administración, pensando no sólo en los intrusos, sino también en una mala práctica por parte de usuarios autorizados.

Para deshabilitar la edición de ficheros desde el panel de administración de WordPress, introduciremos la siguiente línea en nuestro fichero de configuración 'wp-config.php':

```
define( 'DISALLOW_FILE_EDIT', true );
```

Este código equivaldría a eliminar los permisos 'edit\_themes', 'edit\_plugins' y 'edit\_files' de un usuario registrado en el sitio.

Podemos añadir otra capa adicional de control para entornos de producción donde no deseamos que el usuario instale temas o plugins por su cuenta. Para ello, de nuevo añadiremos a nuestro fichero de configuración wp-config.php, la siguiente línea:

```
define( 'DISALLOW_FILE_MODS', true );
```

Recuerda desactivarla, poner la directiva a false, si necesitas realizar tareas de administración sobre dicha instalación.

**Nota:** recuerda que todas las modificaciones que realices al fichero wp-config.php debes realizarlas antes de la siguiente línea:

```
/* ¡Eso es todo, deja de editar! Feliz blogging. */
```

## 6. UTILIZA UNA CDN POR DELANTE



Aunque ya conocemos los beneficios de disponer de un servicio de CDN para mejorar el rendimiento de un proyecto web, el uso de una CDN de tipo DNS (por delante de nuestro servidor web) también mejorará la seguridad de nuestro proyecto en tres aspectos:

- *Disponer de un Firewall activo que actualiza continuamente frente a comportamientos sospechosos hacia sitios web: conexiones masivas, rastreo de puertos, etc.*
- *Previene ante ataques de fuerza bruta. Ya que al utilizar la red distribuida de servidores del proveedor minimiza el impacto y aplica reglas de bloqueo inmediatamente al detectar este tipo de ataques, normalmente DoS o DDoS.*
- *Enmascara la dirección real de nuestra máquina. Previene ataques directos contra nuestro sitio al desconocer la IP real de donde se aloja.*

Te recomiendo utilizar Cloudflare como CDN para mejorar la seguridad y el rendimiento de tu sitio web con WordPress. En SiteGround, todos los planes de alojamiento disponen de una cuenta gratuita de Cloudflare.

## 7. COPIAS DE SEGURIDAD



Aunque no queremos que nunca tengas que recurrir a este truco, siempre es mejor prevenir teniendo una copia de seguridad de nuestro sitio ante un posible incidente grave.

Muy raramente se necesita restaurar una copia de seguridad, pero en SiteGround disponemos de un sistema de backup/restauración creado por nosotros, independiente de la infraestructura de servicio web, para que te sientas tranquilo ante una posible incidencia y no penalice el servicio a tus usuarios.

Mi recomendación, es seguir la regla 3-2-1 como estrategia para tus copias de seguridad que contienen datos importantes, que consiste en:

- *Disponer de 3 copias de los datos.*
- *En 2 formatos diferentes como mínimo.*
- *Con 1 copia en otro lugar físico.*

De nada sirve tener las copias en el mismo soporte o en la misma localización física en caso de desastre.

Y no te olvides, después de cambios importantes en WordPress realizar una copia de seguridad preventiva.



# ASEGURANDO LA PÁGINA DE ACCESO Y SESIONES

## 8. ACTIVAR Y FORZAR HTTPS



El protocolo HTTPS permite la comunicación segura de la información entre el usuario y el servidor, eliminando los posibles ataques “man-in-the-middle” que se producen al colocar servicios intermedios que alteran o adquieren la información entre los dos extremos, encriptando la información sensible.

Para poder utilizar protocolo HTTPS en tu sitio, tienes que instalar un certificado en tu servidor web y cambiar la URL en tu panel de administración.

En SiteGround todos nuestros planes de hosting incluyen certificados Let’s Encrypt de manera gratuita y un sencillo instalador dentro del panel de control para configurarlo. Lo encontrarás en “Seguridad”, “Let’s Encrypt”.

Existen varios plugins de WordPress que te permiten además forzar las conexiones de tu sitio a HTTPS de todos los recursos que utilizas dentro de tu contenido, para evitar avisos o errores relacionados con contenido mixto (servir contenido HTTP y HTTPS en la misma página).

Por último, debes forzar que las sesiones de acceso al panel de administración de tu sitio con WordPress se realicen bajo protocolo SSL, simplemente añadiendo el siguiente código al fichero wp-config.php:

```
define('FORCE_SSL_LOGIN', true);
```

```
define('FORCE_SSL_ADMIN', true);
```

**Nota:** recuerda que antes debes tener un certificado SSL activo en tu instalación, por ejemplo el que proporciona Let’s Encrypt

## 9. DESACTIVAR SUGERENCIAS DE SESIÓN



Como ya hemos comentado, y repetiremos, una de las principales tareas para proteger nuestro sitio de posibles intrusiones, es dar la menor información posible al atacante para facilitarle su trabajo. En este truco te proponemos cómo minimizar las posibles entradas a

nuestro sitio deshabilitando las sugerencias de inicio de sesión del formulario de acceso o registro, que por defecto, indican si el usuario o clave no son correctos.

Para poder deshabilitar esas sugerencias por defecto de WordPress, basta con añadir el siguiente código a tu archivo `functions.php` del tema activo o a tu plugin de utilidades:

```
function no_wordpress_login_errors(){  
    return 'Gracias por intentarlo, pero ya estamos protegidos';  
}  
add_filter( 'login_errors', no_wordpress_login_errors );
```

**Nota:** puedes personalizar el mensaje a tu gusto.

## 10. MOVER EL ACCESO A LA ADMINISTRACIÓN DE TU SITIO



Muchos de los ataques recibidos por sitios se han llevado a cabo mediante bots que identifican que tras esa web se esconde un WordPress, simplemente añadiendo al nombre de dominio la ruta “/wp-admin” y accediendo a la pantalla de registro, donde pueden forzar la entrada de una manera fácil si nuestros nombres de usuario y contraseñas son débiles. Para entendernos, todos los sitios con WordPress tienen la puerta de entrada en el mismo lugar, ¿por qué no cambiarla de sitio para hacer más difícil el acceso?

Existen varios plugins en el repositorio que te permiten realizar este cambio, moviendo la ruta de entrada por una cualquiera, por ejemplo, [www.minombredesitio.com/minuevopaneldeadmin](http://www.minombredesitio.com/minuevopaneldeadmin)

Recomiendo utilizar el plugin WPS Hide Login, aunque puedes utilizar otros, e incluso algún plugin de seguridad de tipo Firewall incluye esta funcionalidad.

## 11. LIMITAR LOS INTENTOS FALLIDOS DE ACCESO



Esta técnica consiste en bloquear el acceso durante unos minutos, o un tiempo más largo o permanentemente, si introducimos usuarios o contraseñas incorrectas un número determinado de veces, dificultando la tarea de los programas automáticos de acceso por fuerza bruta.

Las suites de seguridad como Wordfence, suelen tener estas opciones, pero también podemos implementar el límite de intentos de sesión con plugins

- *Limit Login Attempts (de miniorange)*
- *Limit Login Attempts Reloaded*
- *Loginizer*

Algunos plugins de tipo firewall añaden además esta funcionalidad.

## 12. PLUGIN DE TIPO FIREWALL



Un firewall, es una capa adicional de seguridad software que nos ofrece protección por delante o en nuestra propia instalación, mediante la detección y análisis de las conexiones entrantes. Estas suites de seguridad son muy eficaces y tienen la ventaja de que todo se configura desde un único plugin.

Suelen incluir un Firewall WAF (Web Application Firewall), una herramienta que analiza y bloquea los ataques a nuestra página web en tiempo real. En SiteGround nuestros clientes ya disfrutaban de este servicio, donde analizamos masivamente la tipología de las conexiones y bloqueamos aquellos intentos de ataque, de manera completamente transparente para nuestros clientes.

Algunos de estos plugins son:

- *Wordfence Security (cuidado con la activación de la función Live Traffic, que nos puede dejar sin servicio por sobrecarga del servidor)*
- *All in one security and firewall*
- *iThemes Security (anteriormente Better WP Security)*

Las medidas de seguridad varían un poco según el plugin, pero suelen tener las siguientes características:

- *Escáner de nuestros archivos para buscar cambios, errores y virus*
- *Firewall WAF que detecta y bloquea visitas maliciosas*
- *Visor de tráfico de la web en tiempo real*
- *Herramienta para bloquear el acceso a la web por IP´s*
- *Asegura el Login de WordPress con captcha o límites de intentos de sesión*
- *Auditoria de contraseñas*
- *Verificación en dos pasos para acceder al login de WordPress*
- *Bloqueo por países*

## 13. CABECERAS DE SEGURIDAD



En este truco recopilamos diferentes estrategias para mejorar la seguridad de tu sitio mediante la implementación de una serie de cabeceras que incorporamos en el servidor web y que se envían al navegador.

Comenzamos con la cabecera X-Frame-Options, que nos sirve para prevenir que una de nuestras páginas pueda ser abierta en un frame o iframe externo, ayudándonos a prevenir ataques de tipo clickjacking sobre nuestra web: una técnica para engañar a los usuarios de Internet con el fin de que revelen información confidencial sobre una web aparentemente normal.

Con el siguiente código en nuestro fichero .htaccess, indicamos al navegador que sólo se pueden abrir frames del mismo dominio u origen:

```
Header set X-Frame-Options SAMEORIGIN
```

Si dentro de nuestra web, tenemos servicios que pueden ser embebidos por terceros, podemos especificar a qué dominios damos acceso y prohibir el resto. Por ejemplo:

```
Header set X-Frame-Options "ALLOW-FROM https://example.com/"
```

Continuamos mejorando la seguridad de nuestro sitio frente a ataques XSS (cross-site scripting) en navegadores antiguos. En este caso añadimos la siguiente línea a nuestro fichero .htaccess para que el navegador active la protección por filtrado XSS:

```
Header set X-XSS-Protection "1; mode=block"
```

Para reducir el riesgo de ataques XSS, podemos completar el truco anterior gracias a la cabecera Content-Security-Policy o política de seguridad de contenidos del navegador, que nos permite especificar qué contenidos están permitidos cargar dinámicamente de nuestro sitio o de terceros.

Por ejemplo, si queremos que nuestra página sólo acepte contenidos de nuestro sitio, mismo dominio, debemos añadir la siguiente línea a nuestro archivo .htaccess:

```
Header set Content-Security-Policy "default-src 'self';"
```

Con esta instrucción, bloquearemos la carga de scripts externos a nuestro sitio.

Podemos modificar las variables a nuestro gusto para adaptarlas a nuestro proyecto, por ejemplo si queremos permitir que se carguen scripts de un tercero, como google analytics, la línea quedaría como:

```
header set Content-Security-Policy "script-src 'self' www.google-analytics.com;"
```

Esta cabecera debe implementarse con mucho cuidado, ya que podemos bloquear involuntariamente recursos en nuestro proyecto web, por lo que recomiendo realizar distintas pruebas de esta cabecera sobre un ventana de navegador para comprobar los errores por consola que se puedan generar.

**Nota:** si anteriormente incluías la cabecera X-Content-Security-Policy en tu servidor, utilizada anteriormente y ya obsoleta, debes eliminarla antes de añadir esta, ya que puede generar errores el uso de ambas cabeceras a la vez.

La cuarta cabecera a utilizar en este truco es X-Content-Type-Options, que nos permite protegernos de cargas no deseadas en estilos y scripts cuando no coinciden los tipos MIME esperados con lo que se declaró en la página. Para añadir esta protección, debemos incorporar la siguiente línea a nuestro archivo .htaccess:

```
Header set X-Content-Type-Options "nosniff"
```



## 14. IMPEDIR ATAQUES XML-RPC



El archivo `xmlrpc.php` es utilizado para que algunas aplicaciones y programas puedan comunicarse con WordPress.

Como, por ejemplo, la App de WordPress o los programas de correo como Outlook y Thunderbird que permiten publicar mediante correo electrónico en WordPress. También algunos plugins como Jetpack o Json Api utilizan el archivo XMLRPC para algunas de sus funciones.

Puedes desactivar completamente el acceso al archivo `xmlrpc.php` con reglas en el archivo `.htaccess` o eliminar el archivo XMLRPC directamente si estamos seguros que no lo utilizamos. Para eliminar el acceso vía `.htaccess`, añadiremos estas líneas a nuestro fichero:

```
# proteger el archivo xmlrpc.php
```

```
<Files xmlrpc.php>
```

```
Order Deny,Allow
```

```
Deny from all
```

```
</Files>
```

Y también podemos utilizar plugins como Disable XML-RPC, o el propio plugin iThemes Security, anteriormente citado en el truco 14, que permite también bloquear totalmente XMLRPC.

Para aquellos que necesiten de la funcionalidad de este API, la mejor solución es habilitar su uso sólo para la IP desde donde queremos acceder, bloqueando el acceso para el resto. Para este caso concreto, añadir las siguiente líneas al fichero `.htaccess`, modificando la IP a la que permitimos el acceso:

```
<Files xmlrpc.php>
```

```
order deny, allow
```

```
deny from all
```

```
allow from X.X.X.X
```

```
</Files>
```

## 15. DESACTIVAR LA JSON REST API



Desde la versión 4.4 se incorporó la REST API al core de WordPress, permitiendo a cualquier desarrollador interactuar con el sitio mediante el uso de la misma. Esto ha permitido que WordPress llegue a una mayor cantidad de desarrolladores no familiarizados con el propio WordPress, pero al mismo tiempo, deja una puerta abierta ante posibles ataques a nuestro sitio, sobre todo para ataques tipo DDoS.

Si ninguno de tus plugins, o desarrollos a medida, hacen uso de la REST API, puedes desactivarla fácilmente de tu instalación activa, para ello simplemente añade las siguientes líneas en el archivo `functions.php` de tu tema activo o en tu plugin de recursos:

```
add_filter('json_enabled', '__return_false');
```

```
add_filter('json_jsonp_enabled', '__return_false');
```

Si tienes poca experiencia con el código, también existe un plugin llamado `Disable REST API` que realiza la misma acción de deshabilitar la REST API. También puedes usar el plugin `iThemes Security`, citado en el truco 14, que permite mantener activa la REST API pero con acceso restringido, solo para usuarios con permisos exclusivos.

# 3

## MANTENER LA INSTALACIÓN DE WORDPRESS SEGURA

## 16. PLUGINS Y TEMAS SÓLO DE SITIOS RECONOCIDOS



Plugins y temas son los recursos de terceros más potentes que podemos utilizar para incrementar las funcionalidades de nuestro sitio de entre los miles disponibles, tanto en el repositorio oficial de WordPress como en otros repositorios externos, conocidos o no. Esto puede provocar un problema grave de seguridad, ya que en la mayoría de casos no realizamos un examen exhaustivo del código y funcionalidades que trae consigo ese software que instalamos y activamos, pudiendo provocar brechas de seguridad o funcionamiento no deseado.

Mi recomendación es que sólo descargues plugins y temas del repositorio oficial de WordPress y de sitios conocidos, pero sobre todo, antes de descargar:

- *Mira las opiniones, descargas, comentarios y revisiones existentes.*
- *Desconfía también del software muy antiguo o con pocas actualizaciones.*
- *Revisa quién es el autor y si posee otros en el repositorio.*
- *Si existe compatibilidad sobre el software.*

Y, como siempre, antes de cualquier instalación de un nuevo tema o plugin, realiza una copia de seguridad completa.

## 17. ELIMINA INFORMACIÓN DE VERSIÓN DE WORDPRESS



Una vieja norma de seguridad es dar la menor información posible a los atacantes, y en este caso nos vamos a centrar en la versión de nuestro WordPress, ocultándola de nuestro código HTML que enviamos.

Puedes quitar información de la cabecera HTML y de los ficheros estáticos simplemente añadiendo este código al archivo `functions.php` de tu tema activo o en tu plugin de utilidades:

```
/*
```

```
Ocultar la versión de WP de los scripts y estilos
```

```
*/
```

```
function SG_remove_wp_version_strings( $src ) {
```

```
    global $wp_version;
```

```
    parse_str(parse_url($src, PHP_URL_QUERY), $query);
```

```
    if ( !empty($query['ver']) && $query['ver'] === $wp_version ) {
```

```
        $src = remove_query_arg('ver', $src);
```

```
    }
```

```
    return $src;
```

```
}
```

```
add_filter( 'script_loader_src', 'SG_remove_wp_version_strings' );
```

```
add_filter( 'style_loader_src', 'SG_remove_wp_version_strings' );
```

```
/*
```

```
Ocultar la etiqueta generator de la cabecera
```

```
*/
```

```
function SG_remove_wp_generator() {
```

```
    return "";
```

```
}
```

```
add_filter('the_generator', 'SG_remove_wp_generator');
```

Por otro lado, ocultaremos la información de los propios ficheros de WordPress sobre la versión existente simplemente añadiendo en el fichero `.htaccess` de la raíz de tu WordPress las siguientes líneas:

```
#Bloquear información sobre WP
```

```
<files readme.html>
```

```
Order allow,deny
```

```
Deny from all
```

```
</Files>
```

```
<files license.txt>
```

```
Order allow,deny
```

```
Deny from all
```

```
</files>
```

**Nota:** aunque en algunas guías relacionadas con la seguridad en WordPress se indica que dichos ficheros deben borrarse, mi recomendación es la de bloquear su acceso, ya que es posible que una actualización de WordPress o una reinstalación los descarguen de nuevo.

## 18. DESACTIVAR EL REPORTE DE ERRORES POR PANTALLA



Continuamos con los trucos para reducir la información útil para los atacantes, en este caso, eliminando el reporte de errores por pantalla. De esta manera eliminamos información útil para el atacante como puede ser la versión de PHP o WordPress, la localización de nuestros directorios o información de nuestro servidor.

En entornos de desarrollo, tener habilitado el reporte de errores es muy útil para poder validar nuestro trabajo y encontrar posibles errores, sin embargo, en nuestro entorno de producción debemos desactivar este registro, para que no se muestre la información sobre los posibles errores, donde normalmente aparecen rutas, nombres, versiones, etc.

Para desactivar el reporte de errores en WordPress basta con añadir estas líneas al archivo wp-config.php:

```
error_reporting( 0 );
```

```
ini_set( 'display_errors', 0 );
```

## 19. OCULTAR LA INFORMACIÓN DE APACHE Y DE PHP



Último truco para ocultar información, en este caso sobre las cabeceras que envía nuestro servidor web relacionadas con el software de servidor instalado y la versión de PHP que se ejecuta.

Para ocultar, o limitar dependiendo de la instalación, la información sobre el servidor web, debemos añadir la siguiente línea al fichero .htaccess en el raíz de nuestra instalación:

```
ServerSignature Off
```

Y para ocultar la información sobre nuestra versión de PHP en ejecución, y que algunos servidores la envían en la cabecera HTTP, tenemos dos alternativas. Podemos hacerlo mediante el fichero .htaccess añadiendo la siguiente línea:

```
Header unset X-Powered-By
```

O podemos utilizar la siguiente directiva a nuestro fichero php.ini:

```
expose_php = Off
```

**Nota:** verifica con tu proveedor de hosting cómo puedes añadir esta línea a tu fichero php.ini activo, normalmente se realiza a través del panel de administración del servidor.

## 20. WORDPRESS ACTUALIZADO



Tanto de manera manual, mi recomendación, como de manera automática, la mejor forma de prevenir posibles fallos de seguridad en tu sitio frente a vulnerabilidades detectadas, consiste en tener actualizados los tres bloques principales de tu proyecto web con WordPress: core o núcleo de WordPress, plugins instalados y temas instalados.

En mi caso particular, me gusta realizar esta tarea de manera manual, aunque requiera de más atención y tiempo, y de paso conocer cuales son las novedades que incluye cada actualización y el porqué de algunas. En cuanto al orden de actualización, siempre recomiendo actualizar primero el core y luego, indistintamente, plugins o temas.

Si por el contrario, deseas que el core de WordPress se actualice automáticamente, sólo tienes que añadir el siguiente código en tu archivo wp-config.php:

```
define( 'WP_AUTO_UPDATE_CORE', true );
```

**Nota:** si has desactivado el cron de WordPress no se realizará la actualización automática. Recuerda que recibirás un correo en la cuenta del administrador de la plataforma tras cada actualización realizada.

Y no sólo el core es necesario que esté actualizado, es importante recordar que según un informe de wpscan.org, el 52% de las vulnerabilidades encontradas en instalaciones con WordPress se debían a los plugins, el 11% a los temas y el 37% al propio core, así que si deseas que automáticamente se actualicen los plugins, añade la siguiente línea de código a tu fichero functions.php del tema activo o en tu plugin de funcionalidades:

```
add_filter( 'auto_update_plugin', '__return_true' );
```



Aprovecha antes de añadir este código a eliminar todos los plugins que no estés utilizando en tu sitio y tienes desactivados, no basta con desactivar un plugin para solucionar una posible vulnerabilidad: ¡elimínalo!

Y si también deseas actualizar los temas automáticamente:

```
add_filter( 'auto_update_theme', '__return_true' );
```

Y por último, conviene recordar que no sólo es necesario que nuestro sitio con WordPress esté actualizado, también debería estarlo nuestro ordenador, para que no sea un punto de entrada de software malicioso. Antivirus o actualizaciones del sistema operativo también son trucos de seguridad necesarios.

## 21. HOSTING DE CONFIANZA



Último truco, que en realidad debería ser el primero por importancia, ya que el servidor donde decidimos poner nuestro proyecto web es de suma importancia en lo que a seguridad se refiere.

Tu proveedor de hosting debe ofrecerte una plataforma segura, actualizada y debe estar preocupado por la seguridad de sus infraestructuras y la de sus clientes muy activamente. Desconfía de versiones de software antiguas, accesos no seguros, pocas actualizaciones o desconocimiento de WordPress por parte del soporte técnico.

Elige correctamente tu proveedor de hosting, ya que influirá en gran medida en el éxito y en la seguridad de tu proyecto WordPress.

# CONCLUSIÓN



## RESUMEN

Dicen que el sentido común es el menos común de los sentidos, pero en materia de seguridad es quizá nuestro mejor aliado: usar contraseñas largas, eliminar los usuarios inactivos, política correcta de roles por usuario, no guardar sesiones en ordenadores públicos, mantener la hora del servidor actualizada, acceder desde sitios seguros, monitorizar nuestro sitio...

No hay nada 100% seguro, pero espero que esta guía te permita mantener tu sitio seguro por mucho tiempo. Aplica todos estos consejos con lógica y cabeza, aplicando sólo aquellos que necesites y sean compatibles con tu proyecto, infórmate y mantente actualizado de las novedades en cuanto a seguridad y WordPress.

Consulta a tu proveedor de hosting si tienes duda sobre alguna funcionalidad relacionada con la seguridad, en SiteGround dispones de los mejores profesionales en atención al cliente especializados en WordPress.

Esta guía se basa en mi experiencia durante los más de 10 años que llevo trabajando con WordPress, el conocimiento adquirido gracias a los profesionales de la comunidad, a los que agradezco enormemente su esfuerzo y conocimientos, y la multitud de recursos disponibles acerca de seguridad y WordPress que hay en la red y que se generan casi a diario, por lo que, seguramente, cuando estés leyendo esta guía algunos trucos ya estarán obsoletos y otros no serán necesarios debido a una actualización de WordPress.

Gracias por acompañarme hasta aquí y no dudes en preguntarme vía Twitter en [@fpuenteonline](#)

Un abrazo,  
Fernando Puente




# **SOBRE EL AUTOR**



## FERNANDO PUENTE

Consultor Enterprise en SiteGround  
España

Fernando es un informático de vocación y profesión, formador ocasional y beginner de comer y beber. Tiene más de 21 años de experiencia trabajando en TI, los últimos 12 años en medios de comunicación. Empezó a trabajar con WordPress en 2007 pero no se enamoró de él hasta bien entrada la versión 3. Él está especializado en performance para grandes medios y comercio online. Actualmente, trabaja como consultor de negocio y responsable técnico de diferentes plataformas online.

 [fpunteonline](#)



 SiteGround