WWW.SOLOWORDPRESS.ES - NÚMERO 02 - DICIEMBRE 2019-ENERO 2020

SoloWordpress

TU REVISTA SOBRE WORDPRESS

UN SEO BASICO

WORDPRESS Y LA SEGURIDAD QUE NO PUEDE CONTROLAR

+ ADEMÁS

MANUALES
CONSEJOS
TRUCOS

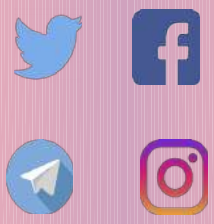
REDACCIÓN:

- Sergio G. B. (Administrador. Redactor artículos SoloLinux)
  info@sololinux.es
- Henry G. R. (Redactor artículos SoloWordPress)
  info@solowordpress.es

MAQUETACIÓN Y EDICIÓN:

- Adrián A. A.
  adrian@sololinux.es

Síguenos en las Redes:



SoloWordPress Magazine esta realizada con **Libre Office Impress 6.2.8**.

AGRADECIMIENTOS:
Diseño de la portada:
Diego Bolivar

CONTACTO:
Para cualquier consulta sobre las revistas, publicidad o colaboraciones escribir un email a:
adrian@sololinux.es

Esta revista se edita gracias a mucha gente y, quiero agradecer a Sergio y a Adrián por su labor.

A Sergio, porque un día hablando sobre cosas de Linux y el Software Libre, va y me suelta, «A mi me gusta como escribes» y luego dice, «¿Y si hacemos algo sobre WordPress?» y yo me crezco y dije «¡Claro, adelante» y empezamos con la web de solowordpress.es.

- Lo bueno: que se ha convertido en un reto seguir la senda de la pagina madre sololinux.es
- Lo malo: que se ha convertido en un reto seguir la senda de la pagina madre sololinux.es

Seguir el ritmo de producción y con la calidad que lo hace Sergio, ¡pufff!
Y un día viene Adrián y me suelta, «Salimos con la revista este viernes; ya tengo todo listo.»
Y yo pienso: ¿Y ahora? ¿no puedo decirles que no?

Tengo que aprovechar mis años de experiencia escribiendo y dando clases en la universidad. Si «lo mio» es la seguridad y el desarrollo, pero no puede ser tan difícil ¿no? … ¡venga, acepto el reto!

Y salimos con la segunda… ¿y cuanto duraremos? Pues hasta que nos digan que ya no nos quieren. Pero mientras seguimos teniendo lectores, seguimos recibiendo aportaciones y colaboraciones… tenemos para mucho rato

Una cosa tengo clara después de tanta experiencia en desarrollo y en tecnología en general: no conozco ningún problema que tenga una única solución. Toda aportación tiene algo bueno y el que un problema se haya resuelto de una manera determinada, no significa que con una nueva visión, una nueva aportación, no podamos conseguir arreglar el problema de una manera nueva.

Así que solo me queda algo por decir:

¡Gracias!
HENRY

PUBLICIDAD

Quieres poner publicidad en la revista, ahora puedes hacerlo de forma muy simple, llegando a todo el mundo con esta revista digital sobre el **CMS** de moda de los últimos tiempos.

CON SOLOWORDPRESS MULTIPLICARAS TUS CLIENTES

Para mayor información escribe un email a: adrian@sololinux.es

Esta revista es de **distribución gratuita**, si lo consideras oportuno puedes ponerle precio.
Tu también puedes ayudar, contamos con la posibilidad de hacer donaciones para la REVISTA, de manera muy simple a través de **PAYPAL**

AYUDANOS A SEGUIR CRECIENDO

COLABORA

Quieres colaborar en la revista.
Para mayor información escribe un email a: adrian@sololinux.es

La **Revista SOLOWORDPRESS**, se distribuye gratuitamente en forma digital para todo el mundo que quiere disfrutar de ella. Si quieres imprimirla es cosa tuya.

PayPal™
Apoyo con una donación

- Revista Mensual
- Noticias
- Manuales
- Distros Linux
- Seguridad
- Redes
- Hardware y Software
- Scripts

¡Si crees que puedes ayudar contacta con nosotros!

SoloLinux

Si el formato digital no te convence, también tenemos todo el contenido en una Página Web

¡Visítanos!

www.sololinux.es

THANKS!

TU PUBLICIDAD AQUÍ

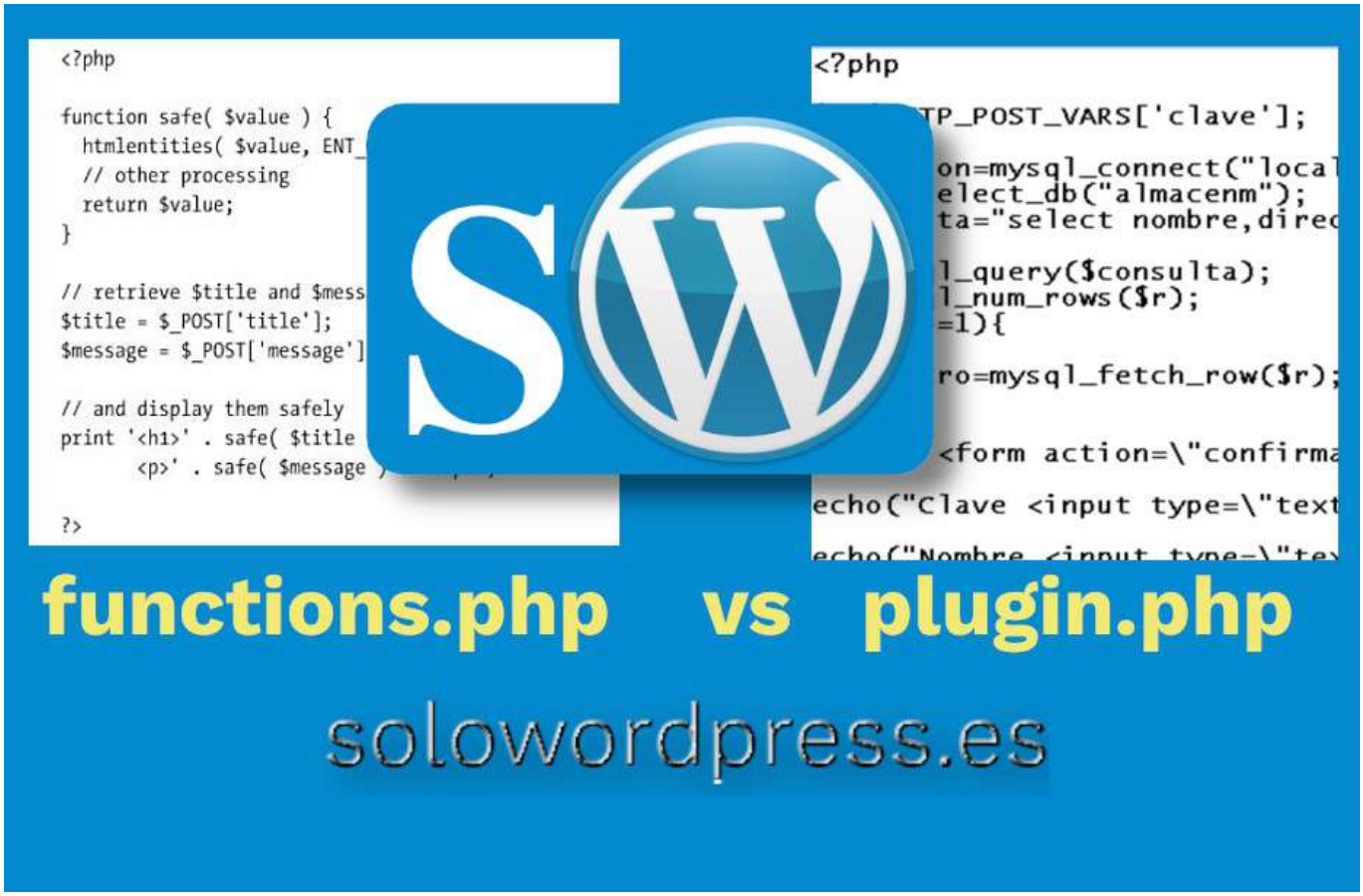
QUIERES APARECER EN LA REVISTA, GANAR CON ELLO MAS VENTAS EN TU WEB, MAS SEGUIDORES EN TUS REDES SOCIALES...

SOLO TIENES QUE MANDAR UN CORREO A

adrian@sololinux.es

Y TE EXPLICAMOS COMO

¿functions.php o plugin?



Es posible (muy probable) que llegues a plantearte si crear un `plugin` o modificar el archivo `functions.php`
Es una pregunta que no tiene una única respuesta, o quizá la respuesta es: «**Depende**».

functions.php o plugin

El archivo `functions.php` no es imprescindible, pero si es muy aconsejable y, cuando te comenté cómo crear un tema hijo, te expliqué cómo crear una versión básica de ese archivo.

Decía también que, «Este archivo es el encargado de ejecutar las funciones internas del tema, sobre los contenidos de WordPress.», pero no sólo vale para eso.

A modo de diferenciación, se puede decir que el archivo `functions.php` se ejecuta sólo cuando está activo el tema que tienes activo (valga la redundancia) mientras que el `plugin` se ejecuta en todo momento, independientemente del tema.

¿Entonces puedo hacer en un plugin lo mismo que en el function.php?

Si, puedes hacer las mismas funciones, pero para hacer las cosas «como es debido», de forma ordenada, debes tener las funciones, filtros y ganchos de uso general, en un `plugin` y las funciones, filtros y ganchos que afectan al tema (a la presentación) en el `functions.php`.

Por ejemplo, si quieres cambiar el orden de los campos de comentarios, al ser esta una función que afecta a la presentación, es decir al tema, debes escribir esa función en el archivo `functions.php`.

Pero si quieres cambiar el comportamiento del editor por defecto, debes hacer esos cambios y situar ese código en el archivo de un `plugin`.

A modo de consejo, asegurate de que no duplicas funciones. Si has creado un plugin y cambias funciones que tenías en el functions.php o viceversa, elimina del primero lo que has pasado al segundo. (Te ahorrarás problemas si actúas con cuidado).

Porqué puedo hacer lo «que me da la gana» con WordPress

WordPress es un CMS, muy extendido, fácil de usar, etcétera.
Una de las razones de su popularidad es, además de lo anterior, que pueden usarlo los desarrolladores y los (permitirme la expresión) simples usuarios; en otras palabras, personas que sólo lo usan en su faceta de «campo de escritura» y personas que construyen sitios web enteros usando WordPress como plataforma.

¿Porqué puedo hacer lo «que me da la gana» con WordPress?
La razón de que pueda hacer uso de WordPress a tu antojo, es que es de «código abierto», lo que permite conocer todo el código interno de la aplicación pero además, lo importante es que este código está disponible bajo la licencia GNUv2.
Esto significa que podemos tomar el código y modificarlo como necesitemos, con ciertas condiciones, no restricciones.
La licencia, que en realidad se llama GNU GPL, siglas en inglés de GNU General Public License o **Licencia Pública General de GNU** fue creada por Richard Stallman, fundador de la « Free Software Foundation (FSF)» para el **proyecto GNU** permite la distribución del software modificado, siempre que se haga bajo la misma licencia.

Vale pero ¿y todo esto, qué significa?
Básicamente, quiere decir que podemos tomar el código de WordPress y adaptarlo a nuestras necesidades. La existencia de Temas y Plugins facilita la tarea, ya que podemos añadir a nuestro sitio, lo que necesitemos para que cumpla nuestros requisitos de presentación y funcionamiento, pero también que las modificaciones las podemos hacer nosotros mismos e incorporarlas a nuestro sitio e incluso distribuirlas a la comunidad de usuarios.

Pero ¿puedo cobrar por mi trabajo?
La licencia GNU no prohíbe cobrar por el trabajo realizado, el echo de que sea «**código abierto**» no significa que no necesite un esfuerzo, un trabajo, un conocimiento; se puede establecer un coste para el producto terminado, como fruto de ese trabajo, pero la licencia en sí determina que los derivados del «*código abierto*» son también (deben ser) «*código abierto*».

Mantenimiento repetitivo

En ocasiones veo…. Aparte del «gracejo», en ocasiones hay que realizar tareas de mantenimiento, como apuntaba en ¿Necesita mantenimiento el Word Press?
y, hay algunas tareas más delicadas que requieren un mayor conocimiento.

El mantenimiento repetitivo
Se trata de tareas esporádicas pero que al ser repetitivas estaría bien poder automatizar. Un ejemplo de esas tareas, sería la «optimización de la base de datos». Es una tarea que debe hacerse pero cuya frecuencia dependerá de muchos factores como la intensidad del uso, el tipo de motor `SQL`, la capacidad del servidor, y muchos más.

Claro que existen plugin que pueden hacer esa tarea por nosotros, pero para quién quiera aprender, contaré cómo se puede realizar. La verdad es que el ejemplo de optimización de la base de datos, quizá no sea el mejor, ya que no se puede establecer a priori cuando será necesario realizar esta tarea, pero vamos allá.

WordPress cuenta con una función llamada cron, que programa las funciones para que se ejecuten a intervalos específicos (diario, semanal, etc.); esto lo vamos a aprovechar para que realice la labor de ejecutar la limpieza.

Como hemos dicho en ¿functions.php o plugin?, podemos insertar estas funciones en cualquiera de ellos; yo recomiendo hacerlo en un plugin.

Empezamos definiendo la tarea que queremos realizar, la función que realizará la optimización y luego, haremos la función que hace uso del cron para programar la periodicidad.

La tarea de optimización
Si fuéramos a realizar esta acción manualmente, deberíamos acceder al administrador SQL de nuestra elección (`phpmyadmin, Webmin, SSMS` …) introducir nuestras credenciales y, seleccionar la base de datos.

Como estamos dentro de WordPress, la base de datos y las tablas están dentro del propio entorno, así que lo único que necesitamos es que la función sepa dónde debe realizar el trabajo.

Para eso, usamos una variable global que WordPress pone a nuestra disposición, se trata de `WPDB`. En realidad, interactuaremos con la clase `$wpdb`.

La clase `$wpdb` tiene varios métodos que podemos ver en el manual (Class Reference/wpdb) pero en esta ocasión sólo nos interesa uno: `get_results`, que con los parámetros «SHOW TABLES» y «ARRAY_A» nos devolverá una matriz asociativa de todos los nombres de tabla en la base de datos.

Con el arreglo obtenido, realizamos un bucle `foreach` recorriendo los valores y realizando la acción de «OPTIMIZE TABLE» en cada uno. Así, la función que necesitamos será:

```php
//Optimize Database
function optimize_database(){
    global $wpdb;
    $all_tables = $wpdb->get_results('SHOW TABLES',ARRAY_A);
    foreach ($all_tables as $tables){
        $table = array_values($tables);
        $wpdb->query("OPTIMIZE TABLE ".$table[0]);
    }
}
```

Hasta aquí todo bien, tenemos ya la función que realizará un recorrido por todas las tablas contenidas en la base de datos e intentará optimizarlas.
Pero esta función no será ejecutada nuca, no tenemos nada que la llame. Es aquí donde necesitamos del cron. Haremos pues, una función que incorpore en las tareas programadas, una «nueva entrada en la agenda». La función que nos permite crear una nueva tarea en la agenda es: `wp_schedule_event()` y haciendo uso de ella, escribimos la función:

```php
function simple_optimization_cron_on(){
    wp_schedule_event(time(), 'daily', 'optimize_database');
}
```

Con esta sencilla función, hemos programado una nueva tarea de ejecución diaria («'daily'») que llama a nuestra función de limpieza, «optimize_database()».

Pero cuidado, esto es inestable, si solo hacemos la inclusión de la tarea, puede llegar a multiplicarse hasta no hacer otra cosa.

Para solucionar este problema, WordPress nos ofrece un conjunto de ganchos específicos, que resuelven este problema. Se trata de las funciones`register_activation_hook` y `register_deactivation_hook`. Estas funciones se invocan cuando un complemento esta activado y desactivado. Nos permitirán hacer que nuestra función sólo se invoque una vez. Para eliminar la función de la agenda, usaremos:

```php
function simple_optimization_cron_off(){
    wp_clear_scheduled_hook('optimize_database');
}
register_activation_hook(__FILE__,'simple_optimization_cron_on');
register_deactivation_hook(__FILE__,'simple_optimization_cron_off');
```

Cómo hacer un plugin en WordPress

Como ya comenté en Mantenimiento Repetitivo, te contaré aquí cómo hacer un plugin en WordPress.

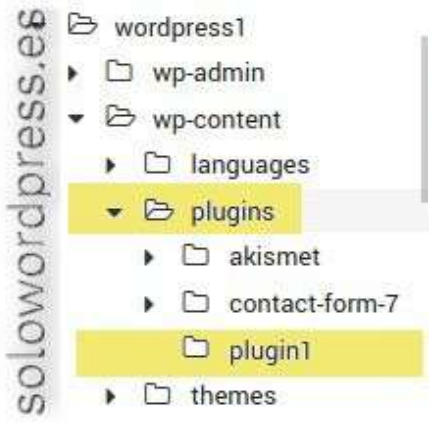
Salvando las distancias, un plugin en WordPress es como una aplicación para un sistema operativo. Ambos son trozos de código, en este caso en lenguaje `php`, que realizan una labor dentro del entorno del sistema que los contiene.

En ambos casos, para que sea posible «instalar» ese código dentro del sistema, se han de cumplir unas reglas y requisitos.

Los requisitos para un `plugin` en WordPress son de fácil cumplimiento, pero conviene conocerlos para que las cosas estén correctamente hechas y podamos aprovechar todas las ventajas que el sistema del entorno nos ofrece. En este artículo te contaré lo mínimo que necesitas para crear un plugin «que funciona»; en otro artículo te contaré las cosas que deberías hacer si quieres hacer un «plugin profesional».

Cómo hacer un plugin en WordPress

Según la estructura de WordPress, un plugin debe estar situado en un lugar determinad, el directorio llamado `plugins`, así que empezaremos por crear un sub directorio, donde estará situado nuestro código.



Hemos llamado `plugin1` a nuestro plugin de ejemplo, pero el nombre puede ser cualquiera, lo importante es que esté situado bajo la carpeta adecuada.
La estructura que ha de tener este subdirectorio, dependerá de su propósito y de lo complicado que sea, así como de las costumbres de cada programador, pero lo único que necesitamos en este directorio, en nuestro caso, será un archivo que llamaremos `plugin1.php`

El archivo que contiene el código de nuestro plugin, sin embargo, si que necesita tener una estructura determinada.
La primera parte del archivo, ha de tener una cabecera informativa, que es la que WordPress usa para identificar el plugin, así que las primeras líneas de código del archivo serán:

```php
<?php
/*
Plugin Name: El nombre que demos al plugin
Plugin URI: Tu pagina web o la del creador
Description: Descripción breve del propósito del plugin
Version: 0.1 (El numero de versión que corresponda)
Author: Tu nombre
Author URI: Tu página web (un poco de publicidad)
License: Lo que corresponda, sugiero: GPLv2 o posterior
.
Cualquier otra nota o explicación acerca del plugin
.
*/

?>
```

Ten en cuenta que el cierre de php no es necesario (la última línea)

Con este contenido, ya es válido para que el plugin sea reconocido por WordPress y lo verás en la lista de plugins si vas al menú `Plugins -> Plugins instalados`.

Claramente, este plugin no hace nada, aunque si queremos, podemos activarlo. Para que el plugin haga algo, debe tener contenido, vamos a agregar el contenido que usamos como ejemplo en el artículo «Mantenimiento repetitivo».

Esto lo hacemos situando el código después del comentario y antes de la línea de cierre de php que, insisto NO es necesaria.

```php
//Optimize Database
function optimize_database(){
    global $wpdb;
    $all_tables = $wpdb->get_results('SHOW TABLES',ARRAY_A);
    foreach ($all_tables as $tables){
        $table = array_values($tables);
        $wpdb->query("OPTIMIZE TABLE ".$table[0]);
    }
}

function simple_optimization_cron_on(){
    wp_schedule_event(time(), 'daily', 'optimize_database');
}

function simple_optimization_cron_off(){
    wp_clear_scheduled_hook('optimize_database');
}

register_activation_hook(__FILE__,'simple_optimization_cron_on');
register_deactivation_hook(__FILE__,'simple_optimization_cron_off');
```

SoloWordpress

Así el contenido completo del archivo `plugin1.php` será:

```php
<?php
/*
Plugin Name: El nombre que demos al plugin
Plugin URI: Tu pagina web o la del creador
Description: Descripción breve del propósito del plugin
Version: 0.1 (El numero de versión que corresponda)
Author: Tu nombre
Author URI: Tu página web (un poco de publicidad)
License: Lo que corresponda, sugiero: GPLv2 o posterior
.
Cualquier otra nota o explicación acerca del plugin
.
*/

//Optimize Database
function optimize_database(){
    global $wpdb;
    $all_tables = $wpdb->get_results('SHOW TABLES',ARRAY_A);
    foreach ($all_tables as $tables){
        $table = array_values($tables);
        $wpdb->query("OPTIMIZE TABLE ".$table[0]);
    }
}

function simple_optimization_cron_on(){
    wp_schedule_event(time(), 'daily', 'optimize_database');
}

function simple_optimization_cron_off(){
    wp_clear_scheduled_hook('optimize_database');
}

register_activation_hook(__FILE__,'simple_optimization_cron_on');
register_deactivation_hook(__FILE__,'simple_optimization_cron_off');
?>
```

Ahora podemos activar nuestro plugin, que realizará su cometido al igual que cualquier otro.

¿Y qué pasa si hay un error?
Si al crear tu plugin has cometido un error de sintaxis o de programación, no te preocupes, no romperás WordPress, el sistema es lo suficientemente inteligente como para detectar la mayoría de errores e ignorar el plugin, indicándolo con un mensaje de error.

Pero no te confíes, los problemas aparecen (como con cualquier aplicación informática) cuando la aplicación no contiene errores de sintaxis ni de estructura, pero no hace lo que pensamos que debería hacer.

La mayoría de las veces, los errores informáticos no significan que el programa no hace lo que le hemos dicho, el programa hace «exactamente» lo que le hemos dicho, que a veces, no es lo que esperamos que haga.

Qué son y para qué sirven los SALT

Es posible que oigas hablar de las SALT de WordPress y te preguntes si es una receta de cocina. Pues no, no tiene nada que ver con la sal en inglés, para decirlo en genérico, un o una SALT es una clave.

Qué son y para qué sirven los SALT

WordPress usa el sistema de SALT para incrementar la seguridad y dar mayor flexibilidad al funcionamiento.
Cuando se accede a WordPress en el lado del «backend» o administrador, mientras estás realizando las tareas oportunas, es necesario saber que sigues conectado. Esto se podría haber hecho con un control de la sesión de php, pero es más seguro y estorba menos si se hace con cookies.
Para mantener la seguridad de las cookies y evitar que si estas son robadas o secuestradas puedan servir para obtener la contraseña del usuario, se usan las claves SALT, de forma que en las cookies no hay elementos fácilmente descifrables.

¿Dónde están las SALT?

Las SALT están definidas en cada instalación de WordPress, en el archivo wp-config.php.
Como ya indiqué en Instalando WordPress en tu servidor lamp, la primera cosa que deberíamos hacer, es proveer las claves de seguridad que usará nuestra copia de WordPress.
Estas claves o valores SALT deben ser únicas en cada instalación.

¿Debo cambiar mis SALT periódicamente?

Yo recomiendo que se cambien, si bien no es necesario hacerlo con periodicidad, si de vez en cuando. Así te aseguras que aún en el caso (difícil) de que alguien pueda secuestrar tus cookies, tendrá menos oportunidades de averiguar tus contraseñas.

¿Cómo cambio las SALT?

Las claves están definidas en el archivo wp-config.php, en el directorio raíz de tu instalación. Debes acceder a ese archivo mediante una cuenta de FTP o SSH (lo que determine tu servicio de alojamiento) y editar el archivo (es un archivo de texto plano).

A continuación localiza un bloque parecido a este:

```
define('AUTH_KEY',         '1jl/vqfs<XhdXoAPz9 ... ... ... c_j{iwqD^<+c9.k<J@4H');
define('SECURE_AUTH_KEY',  'E2N-h2]Dcvp+aS/p7X ... ... ... {Ka(f;rv?Pxf})CgLi-3');
define('LOGGED_IN_KEY',    'W(50,{W^,OPB%PB<JF ... ... ... 2;y&&,2m%3]R6DUth[;88');
define('NONCE_KEY',        'll,4UC)7ua+8<!4VM+ ... ... ... #`DXF+[$atzM7 o^-C7g');
define('AUTH_SALT',        'koMrurzOA+|L_1G}kf ... ... ...  07VC*Lj*1D&&?3w!BT#-');
define('SECURE_AUTH_SALT', 'p32*p,]z%LZ+pAu:VY ... ... ... C-?y+K0DK_+F|0h{!_xY');
define('LOGGED_IN_SALT',   'i^/G2W7!-1H2OQ+t$3 ... ... ... t6**bRVFSD[Hi])-qS`|');
define('NONCE_SALT',       'Q6]U:K?j4L%Z]}h^q7 ... ... ... 1% ^qUswWgn+6&&xqHN&&%');
```

Cambia los valores de la derecha (contenidos entre comillas sencillas ' ').
No es necesario que cambies la cadena de caracteres completa, con que añadas un carácter al principio o al final, o en medio de la cadena, el valor de la clave es completamente distinto.

WordPress 5.3 y la accesibilidad

Desde la creación de WordPress, no se ha tenido en cuenta la accesibilidad. Afortunadamente esto está cambiando y, con cada actualización, se han integrado mejoras en este sentido.

Hay que tener en cuenta que al ser un CMS de uso general, no es fácil tener en cuenta todos los aspectos y, se prima la facilidad de uso, la facilidad de construcción, sobre los detalles «menos importantes».

Pero, vamos por partes.

WordPress 5.3 y la accesibilidad
No vale la pena discutir sobre el pasado, las razones o disculpas de porqué si o porqué no WordPress era accesible. Entre otras razones, porque aunque WordPress fuese 100% accesible desde el principio, quienes hacen desarrollos de temas o plugins para WordPress son los responsables de que el resultado final fuese accesible, al estar en su mano implementar o no las técnicas de accesibilidad correspondientes en cada caso.

Pero es más, suponiendo que WordPress fuese 100% accesible y que los desarrolladores de temas y plugin han hecho su trabajo y el resultado fuese 100% accesible, ya que WordPress es flexible y cada usuario puede crear la combinación que quiera de temas y plugins, al final debe ser el autor, quien se responsabilice de que el producto final sea accesible.

Sea como sea, WordPress en su última versión 5.3 tiene un compromiso con la accesibilidad, como ellos mismos explican en su página sobre accesibilidad.

¿Qué es la accesibilidad en WordPress y porqué me interesa?
WordPress es tan fácil de manejar, que cualquiera puede sucumbir a la tentación de escribir una bitácora. De echo, ese es su objetivo.
Pero en el mundo en el que nos ha tocado vivir, existen personas con unas capacidades que otras no poseen.

Si el mundo fuera perfecto, y los humanos to tuviésemos carencias, todo sería más fácil, un producto hecho para ser usado de una determinada manera, sería usado de esa manera por todos; pero como no es así, hay persona que no pueden usar el producto como se diseñó por que carecen de …
En WordPress se crea contenido, pero al igual que en cualquier página web, ese contenido está pensado para que los visitantes del sitio lo lean.

Por ejemplo, una persona con una ceguera parcial, quizá no puede ver el contenido de tu bitácora, porque el texto es demasiado pequeño.

Una persona con deficiencia total de visión, intentará usar un lector de pantallas para «ver» el contenido de tu bitácora, pero si no está preparada, quizá su experiencia no sea satisfactoria.

En ambos casos, esos visitantes de tu sitio web, preferirán otro sitio al tuyo, si otros le facilitan la tarea. ¿Estás seguro de que puedes «desperdiciar» esas visitas?

Además, ¿has pensado que los «bots» que leen los contenidos en la web, son ciegos?

Cuando las «arañas» de Goggle o Bing o cualquier servicio de indexación visitan tu sitio, no «ven» lo bonito y chulo que está el texto escrito, simplemente ven texto y, si no tienen las indicaciones correctas, se pierden toda la información que «no está clara».

Si el robot abandona tu sitio, ¿en qué posición de «ranking» queda tu sitio?

Así que si eres de los que le interesa el SEO, aunque sólo sea por eso, será mejor que empieces a preocuparte por la accesibilidad.

¿Qué puedo hacer para mejorar la accesibilidad?
Hay muchas cosas que dependen de ti, como ya expliqué en La Tipografía 2, es muy importante tanto el tamaño como la legibilidad de la tipografía que elijas para tu bitácora.

Independientemente de la tipografía elegida por el diseñador del tema que quieras usar, al crear un tema hijo, asegura que defines la fuente que quieres usar.

Además si en tu bitácora pones enlaces a otras páginas, asegúrate de que el enlace está bien creado, es decir, que sea descriptivo o que contenga los atributos `title` y `rel` necesarios.

Si usas palabras que consideras que debes resaltar, no te limites a ponerlas entre comillas, usa los atributos de resaltado, como en, `<strong>palabras resaltadas</strong>`, cuando sea necesario.

Si el texto es muy interesante, pero los colores que usas no tienen suficiente contraste (en relación con el fondo) resultará ilegible para quienes tengan deficiencias de visión o daltonismo.

Otros puntos importantes a citar:
- Utiliza los `alt` en las imágenes.
- Sitúa el contenido relevante al principio.
- Mientras sea posible, no uses `iframe`.
- La accesibilidad no es sinónimo de contenido feo y aburrido.
- Si usas vídeo, pon subtítulos.
- Usa correctamente el marcado de encabezados (h1, h2, h3 …).

Estos son sólo unos cuantos apuntes sobre la accesibilidad, no hay espacio ni es el cometido de esta entrada, extenderme sobre todos los aspectos.

Si crees que es un tema importante para ti y tienes dudas o preguntas o necesitas asesoramiento, contacta con nosotros.

Las constantes de WordPress 5.3

Para los que quieren asumir el control o son desarrolladores en el entorno de WordPress, les conviene conocer las constantes que maneja el CMS.

Como es sabido, en informática las constantes son valores constantes, es decir, que no cambian. Mientras no se definan de nuevo, claro

Las constantes de WordPress
En este entorno, también se usan constantes y, existen métodos que permiten alterar los valores de las constantes, que no veremos aquí para no complicar demasiado.

WordPress hace uso de estas constantes, como casi cualquier otra aplicación informática, para no tener que depender de los «caprichos humanos» y saber en cada momento con qué valores cuenta.

Una constante es un identificador (nombre) para un valor simple. Como el mismo nombre sugiere, ese valor no puede cambiar durante la ejecución de un script. Una constante, por defecto, es sensible a mayúsculas y minúsculas, y por convención los identificadores están siempre en mayúsculas.

En WordPress, podemos decir que hay constantes para muchos usos, tanto que se pueden clasificar y agrupar en 10 categorías.

En este artículo voy a listar todos los grupo, pero no trataré todas las constantes a fondo, simplemente una breve descripción de su utilidad.

Las constantes se agrupan

1. General
2. Estado
3. Rutas, directorios y enlaces
4. Base de datos
5. Multisitio
6. Cache compresión de scripts
7. Sistema de archivos y conexiones
8. Temas
9. Debug
10. Seguridad y cookies

General

- **AUTOSAVE_INTERVAL** Define el intervalo en el que WordPress debería hacer un autoguardado. Valor: tiempo en segundos (Por defecto: 60).
- **CORE_UPGRADE_SKIP_NEW_BUNDLED** Te permite saltar nuevos archivos en paquete como en plugins y/o temas en las actualizaciones. Valores: true | false.
- **DISABLE_WP_CRON** Desactiva la función cron de WordPress. Valor: true
- **EMPTY_TRASH_DAYS** Controla el número de días antes de que WordPress borre permanentemente entradas, páginas, adjuntos y comentarios de la papelera de reciclaje. Valor: tiempo en días (Por defecto: 30).
- **IMAGE_EDIT_OVERWRITE** Permite a WordPress sobreescribir una imagen antes de editar o guardar la imagen como copia. Valores: true | false.
- **MEDIA_TRASH** Activa/Desactiva la función de papelera de reciclaje para los medios. Valores: true | false (Por defecto: false).
- **WPLANG** Define el idioma que usará WordPress en el frontend. Valores: Para Español es_ES.
- **WP_DEFAULT_THEME** Define el tema por defecto para los sitios nuevos, también sirve como respaldo en caso de fallo del tema activo. Valor: nombre del tema (Por defecto: twentytwenty en la 5.3).
- **WP_CRON_LOCK_TIMEOUT** Define un periodo de tiempo en el que se finalizará un único «cronjob». Valor: tiempo en segundos (Por defecto: 60)
- **WP_MAIL_INTERVAL** Define un periodo de tiempo en el que se podrá hacer una única petición de email. Valor: tiempo en segundos (Por defecto: 300).
- **WP_POST_REVISIONS** Activa/desactiva la función de revisión de entradas . Un numero mayor que 0 define el número de revisiones para las entradas. Valores: true | false| número (Por defecto: true)
- **WP_MAX_MEMORY_LIMIT** Te permite cambiar el límite máximo de memoria para algunas funciones de WordPress. Valores: (Por defecto: 256M).
- **WP_MEMORY_LIMIT** Define el límite de memoria para uso de WordPress. Valores: (Por defecto: 32M, para Multisitio 64M)

Estado
La mayoría de estas constantes son definidas en ciertos estados de la ejecución si ocurren ciertas condiciones. Es decir, pueden no existir y solo aparecer en ciertas condiciones.

- **APP_REQUEST** Es definida si hay un Atom Publishing Protocol request. Valor: true.
- **COMMENTS_TEMPLATE** Es definida si se carga el template de comentarios Valor: true.
- **DOING_AJAX** Es definida si hay un request AJAX. Valor: true.
- **DOING_AUTOSAVE** Es definida en el momento en que se esta haciendo un guardado automático de una entrada. Valor: true.
- **DOING_CRON** Es definida si WordPress esta realizando un cronjob. Valor: true.
- **IFRAME_REQUEST** Es definida si hay inline-frame request. Valor: true.
- **IS_PROFILE_PAGE** Es definida si el usuario cambio su perfil. Valor: true.
- **SHORTINIT** Si se define, WordPress cargara lo mínimo indispensable para trabajar. Valor: true.
- **WP_ADMIN** Es definida si hay un request en el backend. Valor: true.
- **WP_BLOG_ADMIN** Es definida si hay un request en /wp-admin/. Valor: true.
- **WP_IMPORTING** Es definida si WordPress esta importando datos. Valor: true.
- **WP_INSTALLING** Es definida si estamos realizando una nueva instalación o actualización. Valor: true.
- **#WP_INSTALLING_NETWORK** Es definida si estamos en el network admin o se esta instalando una red. Valor: true
- **WP_LOAD_IMPORTERS** Es definida si usas Herramientas -> Importar. Valor: true.
- **WP_NETWORK_ADMIN** Es definida si hay un request en /wp-admin/network/. Valor: true.
- **WP_REPAIRING** Es definida si hay un request en /wp-admin/maint/repair.php. Valor: true.
- **WP_SETUP_CONFIG** Es definida mientras WordPress es instalado o configurado. Valor: true.
- **WP_UNINSTALL_PLUGIN** Es definida si un plugin es desinstalado (para uninstall.php). Valor: true.
- **WP_USER_ADMIN** Es definida si hay un request en /wp-admin/user/. Valor: true.
- **XMLRPC_REQUEST** Es definida si hay un request en el API de XML-RPC. Valor: true.

Rutas, directorios y enlaces
Posiblemente las constantes más útiles para todo desarrollador.

- **ABSPATH** Directorio absoluto a la instalación de WordPress. El camino interno del servidor. Default: directorio donde se encuentra wp-load.php.
- **WPINC** Directorio relativo a /wp-includes/. No se puede cambiar. Default: wp-includes.
- **WP_LANG_DIR** Directorio absoluto al directorio con los lenguages. Default: WP_CONTENT_DIR /languages or WP_CONTENT_DIR WPINC /languages.
- **WP_PLUGIN_DIR** Directorio absoluto a la carpeta de plugins. Default: WP_CONTENT_DIR /plugins.
- **WP_PLUGIN_URL** URL a la carpeta de plugins. Default: WP_CONTENT_URL /plugins.
- **WP_CONTENT_DIR** Directorio absoluto a wp-content. Default: ABSPATH wp-content.
- **WP_CONTENT_URL** URL a wp-content. Default: {Site URL}/wp-content.
- **WP_HOME URL** del inicio (Home URL).
- **WP_SITEURL** URL al directorio root de WordPress.
- **WP_TEMP_DIR** Directorio absoluto en dónde se guardan los archivos temporales.
- **WPMU_PLUGIN_DIR** Directorio absoluto al directorio de plugins de la red. (multisitio) Default: WP_CONTENT_DIR /mu-plugins
- **WPMU_PLUGIN_URL** URL al directorio de plugins de la red. (multisitio) Default: WP_CONTENT_URL /mu-plugins.

Base de datos
Lo más importante, que está definido en wp-config.php.

- **DB_CHARSET** Valor: Ver MySQL docs (Default: utf8).
- **DB_COLLATE** Valor: Ver MySQL docs (Default: utf8_general_ci).
- **DB_HOST** Valor: IP address, domain and/or port (Default: localhost).
- **DB_NAME** Valor: Nombre de la base de datos que usamos.
- **DB_PASSWORD** Valor: Contraseña del usuario que usamos.
- **DB_USER** Valor: Nombre del usuario que usamos.
- **WP_ALLOW_REPAIR** Permite automáticamente actualizar y optimizar las tablas de la base de datos /wp-admin/maint/repair.php. Valor: true.
- **CUSTOM_USER_TABLE** Te permite cambiar la tabla de usuarios. Valor: nombre de la nueva tabla.
- C**USTOM_USER_META_TABLE** Te permite cambiar la tabla de meta de usuarios. Valor: nombre de la nueva tabla.

Multisitio

- **ALLOW_SUBDIRECTORY_INSTALL** Te permite instalar Multisitio en un subdirectorio. Valor: true.
- **BLOGUPLOADDIR** Ruta absoluta al directorio de cargas del sitio concreto. Por defecto: WP_CONTENT_DIR /blogs.dir/{Blog ID}/files/
- **BLOG_ID_CURRENT_SITE** ID del blog del sitio principal. Por defecto: 1
- **DOMAIN_CURRENT_SITE** Dominio del sitio principal. Por defecto: dominio
- **DIEONDBERROR** Cuando se define se muestran en pantalla los errores de la base de datos. Valor: true.
- **DIEONDBERROR** Cuando se define se muestran en pantalla los errores de la base de datos. Valor: true.

- **ERRORLOGFILE** Cuando se define se guardan en un archivo de registro los errores de la base de datos. Valor: ruta absoluta a un archivo con permisos de escritura.
- **MULTISITE** Se define si se va a usar Multisitio. Valor: true.
- **NOBLOGREDIRECT** Define una URL de un sitio al que WordPress debería redirigir si está cerrado el registro o un sitio no existe. Valores: %siteurl% para el sitio principal o URL personalizada.
- **PATH_CURRENT_SITE** Ruta al sitio principal.
- **UPLOADBLOGSDIR** Ruta al directorio base de subidas, relativo a ABSPATH. Por defecto: wp-content/blogs.dir
- **SITE_ID_CURRENT_SITE** ID de la red del sitio principal. Por defecto: 1
- **SUBDOMAIN_INSTALL** Define si se instalará un subominio o no. Valores: true | false
- **SUNRISE** Cuando se define WordPres cargará el archivo /wp-content/sunrise.php. Valor: true.
- **UPLOADS** Ruta al directorio de subidas específico de un sitio, relativo a ABSPATH. Por defecto: UPLOADBLOGSDIR /{blogid}/files/
- **WPMU_ACCEL_REDIRECT** Activa/Desactiva soporte para X-Sendfile Header. Valores: true | false (Por defecto: false).
- **WPMU_SENDFILE** Activa/Desactiva soporte para X-Accel-Redirect Header. Valores: true | false (Por defecto: false).
- **WP_ALLOW_MULTISITE** Cuando se define estará disponible la función de Multisitio (Herramientas → Configurar Red). Valor: true.

Cache compresión de scripts

- **WP_CACHE** Cuando se define WordPress cargará el archivo /wp-content/advanced-cache.php. Valores: true | false (Por defecto: false).
- **COMPRESS_CSS** Activa/Desactiva la compresión de las hojas de estilo. Valores: true | false.
- **COMPRESS_SCRIPTS** Activa/Desactiva la compresión de archivos Javascript. Valores: true | false.
- **CONCATENATE_SCRIPTS** Activa/Desactiva la consolidación de archivos CSS y Javascript antes de comprimirlos. Valores: true | false.
- **ENFORCE_GZIP** Activa/Desactiva la salida gzip. Valores: true | false.

Sistema de archivos y conexiones

- **FS_CHMOD_DIR** Define los permisos de lectura y escritura de los directorios. Valores: Ver manual de PHP (Por defecto: 0755).
- **FS_CHMOD_FILE** Define los permisos de lectura y escritura de los archivos. Valores: Ver manual de PHP (Por defecto: 0644).
- **FS_CONNECT_TIMEOUT** Define el tiempo máximo para establecer una conexión. Valores: tiempo en segundos (Por defecto: 30).
- **FS_METHOD** Define el método para conectarse al sistema de archivos. Valores: direct | ssh | ftpext | ftpsockets
- **FS_TIMEOUT** Define el tiempo máximo para una conexión perdida. Valores: tiempo en segundos (Por defecto: 30).
- **FTP_BASE** Ruta al directorio raíz de WordPress. Por defecto: ABSPATH.
- **FTP_CONTENT_DIR** Ruta al directorio /wp-content/. Por defecto: WP_CONTENT_DIR.
- **FTP_HOST** Define el servidor FTP. Valores: Dirección IP, dominio y/o puerto.
- **FTP_LANG_DIR** Ruta al directorio con los archivos del idioma. Por defecto: WP_LANG_DIR.
- **FTP_PASS** Define la contraseña
- **FTP. FTP_PLUGIN_DIR** Ruta al directorio de plugins. Por defecto: WP_PLUGIN_DIR.
- **FTP_PRIKEY** Define una clave privada para SSH.
- **FTP_PUBKEY** Define una clave pública para SSH.
- **FTP_SSH** Activa/Desactiva SSH. Valores: true | false.
- **FTP_SSL** Activa/Desactiva SSL. Valores: true | false.
- **FTP_USER** Define el usuario FTP.
- **WP_PROXY_BYPASS_HOSTS** Te permite definir algunas direcciones que no pasarán por el proxy. Valores: www.ejemplo.com, *.ejemplo.org
- **WP_PROXY_HOST** Define la dirección del proxy. Valores: Dirección IP o dominio
- **WP_PROXY_PASSWORD** Define la contraseña del proxy.
- **WP_PROXY_PORT** Define el puerto del proxy.
- **WP_PROXY_USERNAME** Define el usuario del proxy.
- **WP_HTTP_BLOCK_EXTERNAL** Te permite bloquear peticiones externas. Valores: true | false.
- **WP_ACCESSIBLE_HOSTS** Si se define
- **WP_HTTP_BLOCK_EXTERNAL** puedes añadir servidores que no deberían bloquearse. Valores: www.ejemplo.com, *.ejemplo.org

Temas

- **BACKGROUND_IMAGE** Define una imagen de fondo por defecto.
- **HEADER_IMAGE** Define una imagen de cabecera por defecto.
- **HEADER_IMAGE_HEIGHT** Define la altura de la imagen de cabecera.
- **HEADER_IMAGE_WIDTH** Define el ancho de la imagen de cabecera.
- **HEADER_TEXTCOLOR** Define el color de fuente del texto de la cabecera.
- **NO_HEADER_TEXT** Activa/Desactiva el soporte para texto en la cabecera. Valores: true | false.
- **STYLESHEETPATH** Define la ruta absoluta a la hoja de estilos del tema actual.
- **TEMPLATEPATH** Define la ruta absoluta a los archivos de plantilla del tema actual.
- **WP_USE_THEMES** Activa/Desactiva la activación de temas. Valores: true | false.

Debug

- **SAVEQUERIES** Activa/Desactiva el guardado de las queries de la base de datos en un array ($wpdb->queries). Valores: true | false.
- **SCRIPT_DEBUG** Activa/Desactiva la activación de archivos comprimidos CSS y Javascript. Valores: true | false.
- **WP_DEBUG** Activa/Desactiva el modo debug en WordPress. Valores: true | false (Por defecto: false).
- **WP_DEBUG_DISPLAY** Activa/Desactiva la visualización de errores en pantalla. Valores: true | false | null (Por defecto: true).
- **WP_DEBUG_LOG** Activa/Desactiva la escritura de errores en el archivo /wp-content/debug.log. Valores: true | false (Por defecto: false).

Seguridad y cookies

- **ADMIN_COOKIE_PATH** Ruta al directorio /wp-admin/. Por defecto: SITECOOKIEPATH wp-admin o para Multisitio en subdirectorio SITECOOKIEPATH.
- **ALLOW_UNFILTERED_UPLOADS** Permite subidas sin filtrado para los administradores. Valor: true
- **AUTH_COOKIE** Nombre de la cookie para la identificación. Por defecto: wordpress_ COOKIEHASH
- **AUTH_KEY** Clave secreta. Valores: Ver el generador.
- **AUTH_SALT** Clave secreta. Valores: Ver el generador.
- **COOKIEHASH** Hash para generar nombres de las cookies.
- **COOKIEPATH** Ruta al directorio raíz de WordPress. Por defecto: URL de la portada sin http(s)://.
- **COOKIE_DOMAIN** Dominio de la instalación de WordPress. Por defecto: false o para Multisite con subdominios, dominio el sitio principal.
- **CUSTOM_TAGS** Te permite sobreescribir la lista de etiquetas HTML seguras. Echa un vistazo al archivo /wp-includes/kses.php. Valores: true | false (Por defecto: false).
- **DISALLOW_FILE_EDIT** Te permite desactivar la edición de archivos de temas y plugins con el editor de WordPress. Valor: true.
- **DISALLOW_FILE_MODS** Te permite desactivar la edición, actualización, instalación y borrado de plugins, temas y archivos del núcleo desde el escritorio de WordPress. Valor: true.
- **DISALLOW_UNFILTERED_HTML** Te permite desactivar el HTML sin filtrado para todos los usuarios, administradores incluidos. Valor: true.
- **FORCE_SSL_ADMIN** Activa SSL para los accesos y el escritorio. Valores: true| false (Por defecto: false).
- **FORCE_SSL_LOGIN** Activa SSL para los accesos. Valores: true| false (Por defecto: false).
- **LOGGED_IN_COOKIE** Nombre de la cookie para los accesos. Por defecto: wordpress_logged_in_ COOKIEHASH
- **LOGGED_IN_KEY** Clave secreta. Valores: Ver el generador.
- **LOGGED_IN_SALT** Clave secreta. Valores: Ver el generador.
- **NONCE_KEY** Clave secreta. Valores: Ver el generador.
- **NONCE_SALT** Clave secreta. Valores: Ver el generador.
- **PASS_COOKIE** Nombre de la cookie para la contraseña. Por defecto: wordpresspass_ COOKIEHASH
- **PLUGINS_COOKIE_PATH** Ruta al directorio de plugins. Por defecto: WP_PLUGIN_URL sin http(s)://
- **SECURE_AUTH_COOKIE** Nombre de la cookie para la identificación SSL. Por defecto: wordpress_sec_ COOKIEHASH
- **SECURE_AUTH_KEY** Clave secreta. Valores: Ver el generador.
- **ECURE_AUTH_SALT** Clave secreta. Valores: Ver el generador.
- **SITECOOKIEPATH** Ruta de tu sitio. Por defecto: URL del sitio sin http(s)://
- **TEST_COOKIE** Nombre de la cookie para la cookie de prueba. Por defecto: wordpress_test_cookie.
- **USER_COOKIE** Nombre de la cookie para los usuarios. Por defecto: wordpressuser_ COOKIEHASH

De http a https en WordPress

«A pesar de que hace años (2014) Google recomendó que todos los sitios web mudaran a HTTPS, muchos siguen utilizando el protocolo HTTP. Hasta esa fecha (2014) pocas webs utilizaban HTTPS, parecía un coto exclusivo de sitios con comercio electrónico (tiendas virtuales) y grandes corporaciones; de repente todo cambió.»

Y se da respuesta a preguntas como: «¿Por qué es tan importante cambiar a HTTPS?, ¿realmente vale la pena?, ¿existen diferencias entre HTTP y HTTPS?, ¿afecta al SEO?»

Aconsejo encarecidamente que se lea ese artículo, para despejar dudas. Aquí me centraré en cómo hacer que nuestro sitio con WordPress use esa tecnología.

«La principal diferencia entre HTTP y HTTPS es el certificado SSL.»

Como ya sabemos, WordPress es una aplicación informática, es un CMS, es …

De http a https en WordPress

«¿Cómo funciona HTTPS?: El certificado SSL se encarga de encriptar la información que los usuarios proporcionan al sitio web, básicamente convierte los datos en un código. Si alguien llegara a conseguir esos datos no podría entenderlos debido al cifrado (no hay nada imposible, pero si difícil).»

Como en cualquier aplicación web, para añadir el protocolo https, hay que añadir un certificado SSL.

Muchos de los «paquetes de alojamiento» que ofrecen la instalación de WordPress, ofrecen también la instalación de un certificado SSL, pero en cualquier caso es bueno saber cómo hacerlo nosotros mismos.

«La principal ventaja es la seguridad, pero no es el único beneficio de hacer uso de HTTPS. Cambiar a HTTPS también nos ayuda en nuestro trabajo SEO.»

Requisitos previos

Para instalar un certificado digital SSL en WordPress, necesitas:

- **Adquirir un certificado.**

Una vez cumplidos estos requisitos, pasamos a instalarlo en nuestro servidor de WordPress y esto, podemos hacerlo de dos formas distintas, tu eliges cuál te gusta.

Para instalar un certificado en el servidor, tenga o no WordPress, te sugiero que leas el artículo Instalar SSL certificado seguro Let's Encrypt

Método 1 – Usando un plugin.

Este es el método más sencillo y por tanto, el recomendado a los principiantes.

El plugin que recomiendo, es «Really Simple SSL», no necesitas más que instalarlo y activarlo, él se encargará de realizar las tareas necesarias.

La única intervención por tu parte es, ir al menú Ajustes -> SSL, ahí verás que es plugin te ofrece una serie de consejos y aparecerá un botón «¡Adelante, activa SSL!»

Una vez que pulses el botón, ya tendrás tu WordPress funcionando con seguridad SSL.

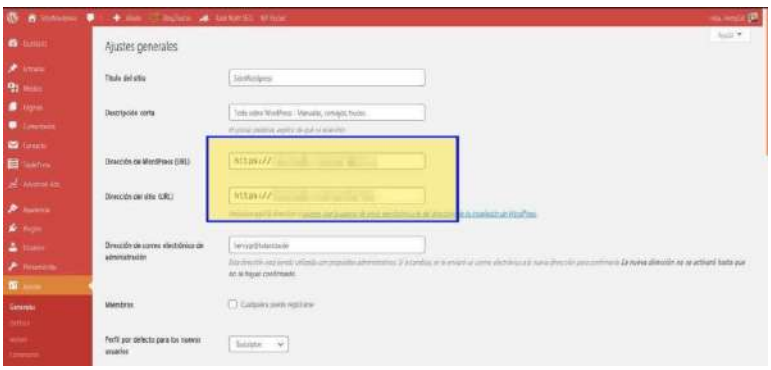
Ten en cuenta que si decides desactivar el plugin, no volverás al estado inicial (aunque ellos lo digan) tendrás una mezcla de las dos tecnologías, lo que es altamente NO recomendable.

Método 2 – Instalando manualmente.

Este método requiere de tu intervención para solucionar varias posibles ejem … … anomalías, cambiando algunos archivos de WordPress.

Como parte de este método, deberás cambiar archivos del tema y del interior de WordPress. Si no te sientes preparado, por favor usa el método 1.

Lo primero es actualizar tu WordPress y los campos de dirección URL, para reflejar que van a ser https://tusitio.xx en lugar de http://tusitio.xx y, esto se hace en el menú Ajustes -> Generales.



No olvides hacer clic en el botón de `Guardar cambios` al final de la página. A continuación, hay que hacer un pequeño cambio en el archivo `.htaccess` para hacer la redirección de `http` a `https`, hay que añadir el código:

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
</IfModule>
```

En el caso en que tengas un servidor con Nginx, deberás usar este código en el archivo de configuración (por supuesto, con el nombre de tu dominio, no ejemplo.com).

```
server {
listen 80;
server_name ejemplo.com www.ejemplo.com;
return 301 https://ejemplo.com$request_uri;
}
```

Si quieres forzar el uso de `https` en la parte de administración, debes cambiar el archivo `wp-config.php` e introducir la siguiente línea de código **antes de la indicación** de «*That's all, stop editing!*»

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

Esto indicará al WordPress que debe forzar `SSL / HTTPS` en el área de administración incluso en los entornos multi sitio.

Una vez que has seguido estos pasos, tu sitio estará completamente listo para usar `SSL / HTTPS`, sin embargo, puede que te encuentres con mensajes de error

La mayoría de los navegadores modernos avisan con ciertos iconos, cuando se accede a sitios considerados seguros o no.

En el caso de los sitios seguros, suele aparecer un candado verde, lo que corresponde a un sitio usando `SSL / HTTPS` y, es posible, que cuando acceden a tu sitio, aparezca un escudo en blanco y negro. Si sitúas el puntero sobre el escudo, te dirá que el sitio contiene partes no seguras.

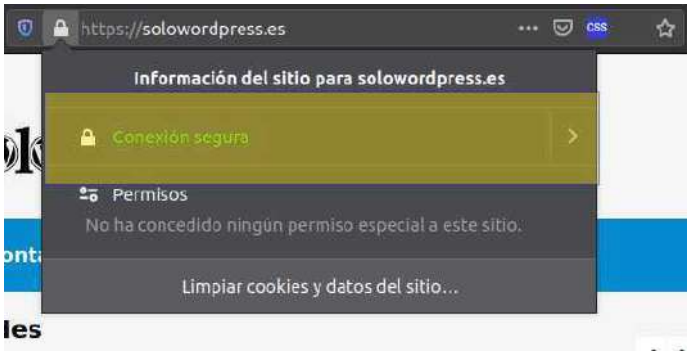
Ese error ocurre cuando hay partes del sitio usando **«https://»** y partes usando **«http://»**

Las Señales de Seguridad

En el artículo De http a https en WordPress comentaba que cuando se usa un certificado SSL, «suele aparecer un candado verde».

Esta afirmación es muy genérica, si bien algunos navegadores como Opera presentan ese candado en verde, otros como el Mozilla Firefox, presentan el candado en color gris cuando la conexión es segura, y el candado en gris con una línea roja, cuando no es segura completamente, presenta «contenido mixto».

Las señales de seguridad
Lo importante es que conozcamos los signos que ofrece el navegador que usemos, no debemos depender de si el candado es de color verde, gris o naranja.
Una de las acciones a tomar (y esto si que funciona en todos los navegadores) es hacer clic sobre el candado (o pulsar en los dispositivos móviles) con lo que se desplegará la información de seguridad de la conexión.
Cuando la conexión es segura, veremos una indicación escrita, generalmente en color verde, que dice: «Conexión segura».



Además, como medida de mayor precaución, debemos fijarnos en que la dirección `URL` empieza con `https://`, como en la imagen.

Si has seguido la guía para convertir to WordPress en seguro e instalar SSL, sabes que en algunos casos, aunque el sitio se acceda mediante el protocolo `https://`, hay contenido mixto, lo que hace que el sitio no sea «totalmente seguro».

Qué es el contenido mixto
El contenido mixto se produce cuando hemos instalado un certificado SSL pero el sitio no está preparado para ello.

Ese contenido se produce cuando partes del código `HTML` del sitio se transmite con protocolo `https` y, partes se transmiten bajo `http`.

Hoy en día, la gran mayoría de temas y plugins para WordPress, están preparados para trabajar con SSL, por lo que casi con seguridad, podemos descartarlos como generadores del problema.

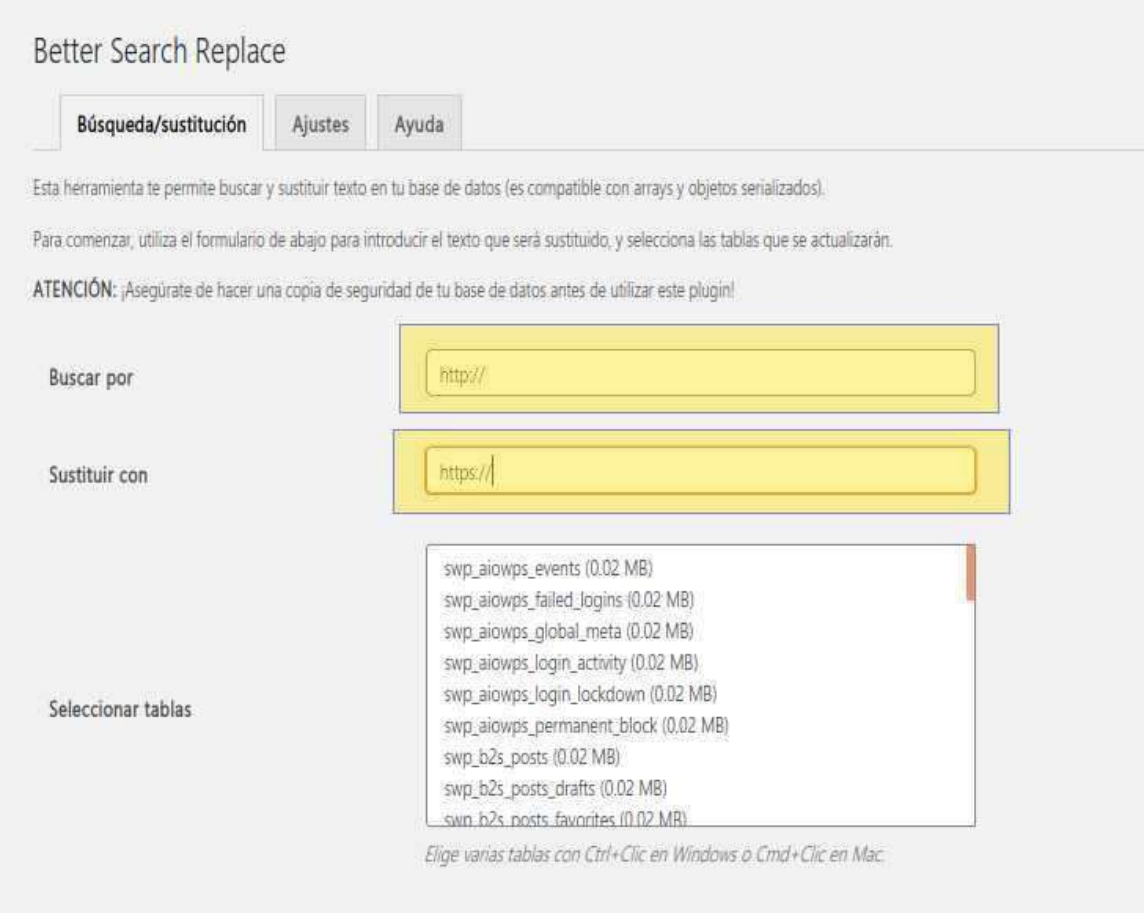
Así que lo más probable es que sea el contenido de nuestra bitácora. Si hemos insertado una imagen en una de las entradas, pero en el momento de la creación no teníamos SSL implementado, es casi seguro que esa imagen tenga como URL, `http://misitio.com/imagen.jpg`, por ejemplo.

Cómo corregir el contenido mixto
De nuevo, la forma más fácil es la instalación de un plugin que haga la tarea.

Lo que hay que hacer es cambiar la base de datos para que las llamadas a imágenes, scripts y otros archivos como los `css`, se hagan con `https://` en lugar del `http://` original.

Un plugin que puede ayudarnos en esa tarea es: «Better Search Replace (https://es.wordpress.org/plugins/better-search-replace/)» que te permite hacer cambios en la base de datos de forma fácil y cómoda.

Una vez instalado y activado este plugin, aparecerá una nueva entrada en el menú `Herramientas` con el nombre de `Better Search Replace`, al pulsar este enlace, aparece la página en la que debemos entrar el texto a buscar en el campo «Buscar por» (http://) y el texto a reemplazar en el campo «Sustituir por» (https://).



Debemos seleccionar las tablas de la base de datos donde queremos hacer el cambio. Si son las entradas, las tablas serán la que hagan alusión a «posts» (en la imagen, swp_b2s_post, swp_b2s_post_drafts, swp_b2s_post_favorities, swp_b2s_post_network_details, swp_b2s_post_sched_details).

Para estar seguro, selecciona todas las tablas de WordPress, ya que puede haber también llamadas http en la biblioteca de medios, por ejemplo.

Asegurate de quitar la marca en la caja de «¿Quieres ejecutar un simulacro?» y pulsa el botón de «Ejecutar búsqueda/sustitución»

Y con esto, ya tienes actualizado tu WordPress, No deberías tener más contenido mixto. Si lo tienes, deberás repasar el tema y los plugin que usas, pero insisto, los temas y plugin modernos ya deberían estar preparados.

Nueva actualización 5.3.1

La mayoría de los usuarios de WordPress nos hemos levantado con la sorpresa de que hay una nueva actualización que, si no has desactivado, se ha producido automáticamente.

Nueva actualización 5.3.1
Se actualiza el motor de WordPress a la versión 5.3.1 Escritorio de actualización Esta versión de seguridad y mantenimiento incluye 46 correcciones y mejoras.

Las correcciones de seguridad solucionan vulnerabilidades referentes a:
- Un usuario no privilegiado podría enviar un mensaje a través de la API REST.
- Cross-site scripting (XSS) que podría ser almacenado en enlaces bien creados.
- Realizar hardening en wp_kses_bad_protocol() para asegurarse de que reconoce el atributo «dos puntos».
- XSS persistente usando contenido del editor de bloques.

También hay correcciones en diseño, WordPress 5.3.1 añade herramientas aún más robustas para crear diseños sorprendentes.
- El nuevo bloque de grupo te permite dividir fácilmente tu página en coloridas secciones
- El bloque de columnas ahora es compatible con anchos de columna fijos.
- Los nuevos diseños predefinidos hacen que sea fácil organizar el contenido en diseños avanzados.
- Ahora los bloques de encabezado ofrecen controles para el color del texto.
- Las opciones de estilo adicionales te permiten establecer tu estilo preferido para cualquier bloque que sea compatible con esta característica.

Si estás interesado en más detalles, tienes una mayor información en las notas de la versión.

WordPress y la Seguridad que no puede controlar

Siempre que hablamos de seguridad en informática, se piensa rápidamente en «virus» y eso está bien, pero hay otras cosas.

Algunos hemos tenido experiencias con los llamados «Troyanos» o con los «Gusanos» y, algunos incluso con los «Ransomware» que les impiden usar su ordenador personal.

WordPress y la Seguridad que no puede controlar
Claro que cuando hablamos de WordPress, a nadie se le ocurre pensar en los «Virus», «Troyanos» o «Gusanos».
Pero WordPress también tiene sus peligros de seguridad aunque, es «más fácil» atajarlos. Como ya dije en la lista de plugins necesarios, hay plugins que se encargan de la seguridad del sitio, cierran «agujeros» que los malos podrían aprovechar, protegen la integridad del sitio, etcétera.

Pero hay un factor que ningún plugin de seguridad puede atajar. Ese factor es tanto o más importante que los demás, ese factor eres tú.

Es probable que nunca te hablasen de la seguridad relacionada con la «ingeniería social».

La ingeniería social se basa en un principio muy básico: "el usuario es el eslabón más débil". A partir de esta idea, busca explotarlo apelando a sus motivaciones más personales, con el objetivo de conseguir que el usuario revele cierta información o permita tomar el control de su equipo. Nuestra mejor defensa es no dejarnos engañar y conocer cómo funcionan este tipo de fraudes y engaños.

Para no complicar mucho el asunto, voy a intentar contarlo de forma que se entienda.

Si tu eres el administrador del servidor en el que se aloja tu WordPress, y entonces es posible que ya sepas de qué te hablo, pero conviene recordar que si te engañan para que facilites tus credenciales (usuario y contraseña) has perdido el control de tu sitio.

Si consiguen engañarte para obtener solo uno, el otro lo descubrirán tarde o temprano.

Si no eres el administrador, sino un simple usuario pero administras la copia de WordPress, estamos en la misma situación, aunque en este caso no podrán usar el servidor entero para sus fines, sino sólo tu WordPress.

Pero claro, siempre es natural pensar «yo no tengo nada interesante, a mí no me atañe», o algo como «si a mí me leen dos gatos, no van a por mí»; siento decírtelo, te equivocas.

Incluso en el caso en que consigan entrar y tú no te percates de ello, veas tu página y asegures que nada ha cambiado.
Los usos que pueden darse a un servidor comprometido, son muchos y variados. Por poner unos ejemplos, enviar correos maliciosos; usar la potencia del servidor para la «criptominería»; usar el servidor como almacenamiento de documentos o imágenes ilegales.

Puedes tener un problema legal (porque tu eres el responsable legal de las malas acciones cometidas con tu servidor).

Consejos
- Cambia con frecuencia tus contraseñas de acceso.
- Siempre que sea posible, utiliza sistemas de «Autenticación en Dos Factores».

Cómo poner código en la cabecera en WordPress

Dejame adivinar, quieres controlar cuantas visitas tienes y el sistema de medición te ha dicho que tienes que poner un código en la cabecera de tu página, pero … ¿Cómo? ¿Dónde?

Si estás leyendo esto es porque tienes una bitácora con WordPress y no sabes qué hacer.
Te daré varias opciones, escoge tu la que te guste. Cuando te han pedido que pongas el código en la cabecera de la página, se refieren a que lo sitúes entre las etiquetas `html: <head> y </head>`.

Si es un sitio elaborado con `html` desde cero, lo tienes fácil, sólo has de introducir ese código en todas las páginas que forman tu sitio.

WordPress es otro mundo, porque cada página se genera dinámicamente con el contenido de cada entrada, así que ¿cómo se hace?

Cómo poner código en la cabecera en WordPress

Primera opción

Como siempre, la opción más fácil es la de usar un plugin que haga el trabajo.
Un plugin como «Insert headers and footers» (https://es.wordpress.org/plugins/insert-headers-and-footers/) te permitirá poner el código y olvidarte de complicaciones técnicas.

Segunda opción

Si has llegado aquí es porque quieres ir más allá, quieres «remangarte y trabajar». ¡Bien!

Si tu tema lo permite, tendrás una opción en el menú de `Apariencia -> Personalizar` en el que te indicarán que puedes «incrustar» código o scripts en la cabecera.

Tercera opción

La fórmula del «DIY» o Hazlo Tu Mismo. Se trata de incrustar el código en las cabeceras ¿no? Pues bien, podemos hacerlo a mano, página a página editando el HTML resultante (lo cual es ilógico, incómodo e inútil) o, podemos pedir a WordPress que lo haga por nosotros, lo que parece una mejor idea.

A WordPress podemos decirle que realice el trabajo, de varias formas. Como es lógico, vamos a decirle (una vez) que lo haga en cada ocasión (todas las veces que construye la página html).

Y aún así, tenemos diferentes formas de hacerlo. Una es cambiar el `script php` que crea las cabeceras html.

El script que hace esa labor se llama `header.php` y si lo modificamos, podemos añadir el código que queramos que aparezca en cada página.

Esto tiene grandes inconvenientes, como:

- Existe un «header.php» en cada tema, por lo que podríamos tener varios, uno por cada tema instalado.
- Si se actualiza el tema, perdemos los cambios hechos.
- Nadie nos garantiza que el tema no modifique este archivo, desde otro script.

Por tanto, nos queda otra opción más interesante, aprovechar las facilidades que nos da WordPress.
Para esto vamos al archivo `functions.php` de nuestro tema hijo. ¡Ah! ¡Que no tienes un tema hijo! Pues debes crear uno. Sigue nuestras indicaciones. Cuando lo tengas creado, seguimos.

En el archivo `functions.php` de nuestro tema hijo, insertamos un «hook» o gancho. En el archivo insertamos el código:

```
function mi_codigo() {
  echo 'Aquí tu contenido';
}
add_action( 'wp_head', 'mi_codigo', 10 );
```

Por supuesto, debes reemplazar «'*Aquí tu contenido*'» con lo que te han indicado que debes insertar en la cabecera de tus páginas, para que funcione tu sistema de medición y seguimiento.

De esta forma, cada vez que WordPress genere una página html, usando el contenido de tu bitácora, incluirá en la cabecera, el código que le has indicado.

Así que, asegurate de que el código html es correcto, si no quieres que tus visitantes tengan sorpresas.

Cómo poner código al final de NO todas las entradas en WordPress

Cómo poner código al final de las entradas en WordPress Una vez más, vamos a usar nuestro archivo de `functions.php` de nuestro tema hijo.

¡Ah! ¡Que no tienes un tema hijo! Pues debes crear uno. Sigue nuestras indicaciones. Cuando lo tengas creado, seguimos.

En el archivo `functions.php` tenemos que insertar una función que haga uso de un gancho de WordPress, en esta ocasión, un gancho de tipo filtro.

El filtro que necesitamos, es uno que nos permita manejar el contenido de cada entrada, así que será uno que maneje «*the_content*

```
// define la llamada a the_content
function filtra_el_contenido( $tras_el_contenido ) {
        // haz aquí tu magia...
        return $tras_el_contenido;
};
// añade el filtro
add_filter( 'the_content', 'filtra_el_contenido', 10, 1 );
```

Realmente, no hay mucho que hacer, sólo debemos añadir después del contenido ya existente, el texto que queramos.

Supongamos que queremos que todas las entradas de nuestra bitácora muestren nuestra admiración por las espinacas, pues podemos añadir la frase: «Y recuerda, un plato de espinacas al día hará que seas el más fuerte.» (Recordando a cierto marinero)

El código que debemos incorporar es tan sencillo como «concatenar» nuestro texto al contenido que va a devolver la función en la sentencia «return». Así, el código sería (ojo a la línea 4):

```
// define la llamada a the_content
function filtra_el_contenido( $tras_el_contenido ) {
        // haz aquí tu magia...
        return $tras_el_contenido . "Y recuerda, un plato
        de espinacas al día hará que seas el mas fuerte.";
};
 // añade el filtro
add_filter( 'the_content', 'filtra_el_contenido', 10, 1 );
```

Por supuesto, dejo a tu imaginación si quieres adornar el texto, recuerda que estas trabajando con una cadena html, por lo que la línea 4 podría ser algo como:

```
return $tras_el_contenido . " Y recuerda, un plato de
    espinacas al día hará que seas el mas fuerte.";
```

Pero lo que queríamos es que no se cumpliese siempre …

Cómo poner código al final de NO todas las entradas en WordPress

Ya que estamos programando y, ya que sabemos cómo hacer que se cumpla siempre, vamos a hacer que se cumpla sólo cuando nos interesa.

Voy a elucubrar y pensar que tienes un blog sobre recetas de cocina.

En ese blog hay algunas entradas con recetas para navidad y claro, no es cuestión de recorrer todas las recetas para ver dónde insertar el código.

Como eres una persona ordenada, tienes categorías para cada caso y, una de las categorías es «receta de navidad».

Y como esto no es una clase de programación, no me extenderé en poner todo el código, sino que te recuerdo que el código añadido puede ser similar a esto:

```
// define la llamada a the_content
function filtra_el_contenido( $tras_el_contenido ) {
        if ($categoria != "receta de navidad"){
                return $tras_el_contenido;
        }
        return $tras_el_contenido . " No olvides que
        tenemos muchas recetas. ¡Feliz Navidad!";
};

// añade el filtro
add_filter( 'the_content', 'filtra_el_contenido', 10, 1 )
```

Así que… ¡La imaginación al poder!

Actualización en WordPress 5.3.2 ¿Defectuosa?

Hoy 19 de diciembre, nos hemos encontrado con una actualización de WordPress, la 5.3.2

Muchas personas se han encontrado con que no pueden crear nuevas entradas ni editar antiguas.

Actualización en WordPress 5.3.2 ¿Defectuosa?
La primera reacción de muchos ha sido de sorpresa y enfado. ¿Cómo es que una actualización pierde todo mi trabajo?

Como con casi todos los problemas informáticos, lo primero es: «Mantén la calma».

Las investigaciones que hemos podido realizar, llevan a que el culpable es un plugin.

Si eres de los que prefieren el editor clásico al editor de bloques, probablemente te has encontrado en esta situación.

La Solución
Nuestras pruebas han llevado a dos posibles soluciones:

- Instala el plugin `TinyMCE Advanced` (https://es.wordpress.org/plugins/tinymce-advanced/).
- Desactiva, borra y reinstala el plugin `Editor Clásico` (https://es.wordpress.org/plugins/classic-editor/).

Cualquiera de estas acciones restaurará las funciones de tu editor clásico.

Además, si encuentras algún plugin que sabes que está instalado y no funciona como debería, simplemente reinstalalo o vuelve a guardar sus ajustes. ¡Feliz edición!

¡Feliz edición!

Actualización en WordPress 5.3.2 ¡Más sorpresas!

Actualización en WordPress 5.3.2 ¡Más sorpresas!
Muchas personas tienen deshabilitadas las actualizaciones automáticas, como ya comenté en actualizaciones automáticas, pero aún así, se han encontrado con la sorpresa de que esta actualización (que además es menor) se ha instalado automáticamente.
Esto está basado en la experiencia de una persona de los que podemos llamar «Gurú», así que cuenta con toda mi confianza.

Cuando se dio cuenta de que había cambiado la versión (porque experimentaba los problemas que ya comenté) empezó a investigar y descubrió un regalito no deseado.

Me extraño mucho que se actualizara wordpress automáticamente, pues yo siempre lo deshabilito y solo actualizo cuando sale la versión en español, primero sale la versión inglesa.

Le había cambiado el archivo `wp-config.php`, incluyendo la siguiente primera línea:
define('WP_AUTO_UPDATE_CORE', 'minor');// Esta opción es imprescindible para garantizar que las actualizaciones de WordPress pueden gestionarse correctamente en el paquete de herramientas de WordPress. Si este sitio web WordPress ya no está gestionado por el paquete de herramientas de WordPress, elimine esta línea.

Los que tengan el archivo en inglés verán:
define('WP_AUTO_UPDATE_CORE', 'minor');// This setting is required to make sure that WordPress updates can be properly managed in WordPress Toolkit. Remove this line if this WordPress website is not managed by WordPress Toolkit anymore.

Este añadido lo ha hecho un elemento externo, en este caso, una herramienta llamada «WordPress Toolkit» que es parte del sistema de administración de servidores llamado «Plesk»

Si no usas esa tecnología, es posible que no te encuentres este inconveniente.
Si recuerdas el artículo sobre «actualizaciones automáticas», te explique que la constante WP_AUTO_UPDATE_CORE puede adquirir varios valores y, en este caso, le han asignado «minor» es decir, se auto actualizarán las versiones menores (p.e. de la 5.3.1 a 5.3.2).

Así que si no quieres actualizaciones automáticas, recuerda cambiar esa línea. Si no tienes una línea como esa, asegurate de añadir una línea de desactivación siempre antes de `/* That's all, stop editing! Happy publishing. */:`
define('WP_AUTO_UPDATE_CORE', false);
define('AUTOMATIC_UPDATER_DISABLED', true);

¡Feliz edición!

Cómo poner texto al final de NO todas las entradas (Nivel avanzado)

Como prometí, hoy os cuento cómo poner texto al final de NO todas las entradas (Nivel avanzado); es la segunda parte de Cómo poner código al final de NO todas las entradas en WordPress.

Por supuesto, podemos añadir texto, código o cualquier otro elemento que seamos capaces de representar con `html`.
Aprovecho para recordar que esto es nivel avanzado, ya que tocaremos codificación `html` y `php`. Así que, vamos allá.

Cómo poner texto al final de NO todas las entradas (Nivel avanzado)
Antes de seguir, ¿has creado ya tu tema hijo? Si no es así, recuerda que tienes las instrucciones en: Cómo crear un tema hijo en WordPress.

Bien, ahora que ya lo tienes, podemos continuar. Lo primero que hemos de localizar, es dónde queremos poner nuestro código. Debido a la estructura de WordPress, debe existir un archivo (script php) que controla la presentación de las entradas.
En este ejemplo, vamos a trabajar con un tema hijo del tema «Twentytwenty» que es el que viene pre configurado con la versión 5.3
En este tema, el archivo encargado es el `/wp-content/themes/twentytwenty/singular.php` y, al editar este archivo, vemos que el bucle que ejecuta para la presentación, llama a otro archivo del tema,

```
while ( have_posts() ) {
	the_post();

	get_template_part( 'template-parts/content', get_post_type() );
}
```

Así que toca editar ese archivo. Pero cuidado, vamos a trabajar en nuestro tema hijo, así que no debemos tocar los archivos del tema padre, lo que tenemos que hacer es copiar la estructura de directorios y editar la copia.

A ver, me explico, como el fichero que queremos modificar se encuentra en:
`/wp-content/themes/twentytwenty/template-parts/content.php`, debemos crear el directorio `/template-parts/` y copiar allí el archivo `content.php`; de modo que tenemos un archivo que editaremos y cuya ruta o camino es: `/wp-content/themes/tema-hijo/template-parts/content.php`.

**** Recuerda ** Al hacer un tema hijo, hay que mantener la estructura de directorios del tema padre, aunque no estén todos los archivos.**

Una vez que tenemos el archivo, lo editamos para localizar dónde debemos incorporar nuestro código. En este caso, el `content.php` tiene a partir de la línea 30:

```
<div class="entry-content">
	<?php
	if ( is_search() || ! is_singular() && 'summary' === get_theme_mod( 'blog_content', 'full' )){
		the_excerpt();
	} else {
		the_content( __( 'Continue reading', 'twentytwenty' ) );
	}
	?>
</div><!-- .entry-content -->
```

Que es efectivamente, el elemento «`div` » donde se escribe el contenido de la entrada, por lo que nuestro código debe ir a continuación.

Nuestro código debe escribir un texto, siempre que se cumpla la condición de que la entrada tiene como categoría, «receta de navidad».
Aprovechamos un código de WordPress para ver las categorías que tiene la entrada y así actuar en caso de coincidencia.
La función `get_the_category()` nos devuelve un arreglo con todas las categorías a las que pertenece una entrada. De ese arreglo, sacamos los nombres de las categorías y comparamos.

```
foreach((get_the_category()) as $category) {
	$nombre_categoria = get_cat_name( $category->cat_ID );
	if ($nombre_categoria == "receta de navidad") {
		echo '<p>¡Feliz Navidad!';
	}
}
```

Como solo puede aparecer una vez el nombre de la categoría que buscamos, no hace falta hacer comprobaciones para evitar duplicados (Si tenemos bien hechas las categorías).

Así que el archivo `content.php` quedaría así:

```
<div class="entry-content">
	<?php
	if ( is_search() || ! is_singular() && 'summary' === get_theme_mod( 'blog_content', 'full' ) ) {
		the_excerpt();
	} else {
		the_content( __( 'Continue reading', 'twentytwenty' ) );
	}
	foreach((get_the_category()) as $category) {
		$nombre_categoria = get_cat_name( $category->cat_ID );
		if ($nombre_categoria == "receta de navidad") {
			echo '<p>¡Feliz Navidad!';
		}
	}
	?>
</div><!-- .entry-content -->
```

Esto es sólo un ejemplo como dije, ¡**la imaginación al poder**!

WordPress no me permite subir imágenes webp.

`WEBP` es un formato de imagen o, mejor dicho, es un formato de archivo de imagen, creado por Google. Está diseñado para ser ligero y por tanto de carga rápida en los navegadores web.

Hoy es un formato «nativo» en muchos navegadores modernos. La ventaja de la forma de codificación de los archivos `WEBP`, hacen que el resultado sea un 28% más pequeño que la misma imagen en un archivo `PNG`.

Así que ¿a qué esperamos? Vamos a implementar todas las imágenes de nuestro sitio en formato `WEBP`, irá todo más rápido. Lo primero que viene a la cabeza es ¿Cómo edito imágenes en ese formato? Respuesta: los programas de tratamiento de imagen como `Photoshop` y `GIMP` soportan ese formato.

Claro, pero ¿Qué hago con las imágenes que ya tengo, cómo las convierto? Respuesta: Google ha previsto esta incidencia y suministra una aplicación para la conversión y hay muchos recursos online para realizar esa conversión (busca por ejemplo: «conversión webp»).

WordPress no me permite subir imágenes webp.
Y cuando ya tenemos todo listo, nos encontramos con que WordPress no nos permite subir ese tipo de imágenes a nuestra biblioteca de medios. Lo siento, este tipo de archivo no está permitido por motivos de seguridad. Por suerte, tiene solución (varias, de echo) como casi siempre. La primera y, como siempre la más fácil, es instalar un plugin que nos permita realizar la tarea, pero… primero debemos saber ¿qué problema queremos solucionar?

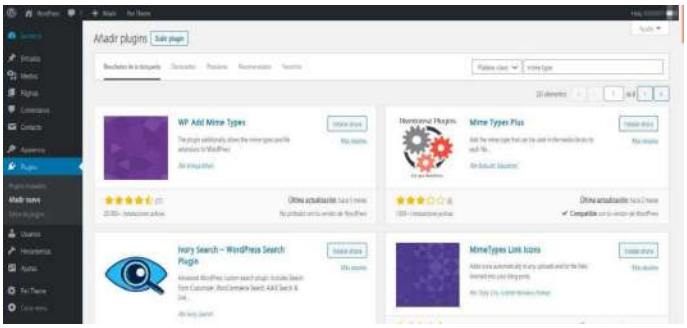
Cuando queremos subir un archivo a la biblioteca, WordPress necesita saber qué tipo de archivo es, para almacenarlo correctamente y, más importante, para tratarlo correctamente durante la «subida» y durante la «manipulación». La diferencia se identifica con lo que se conoce como «`tipo MIME`» y WordPress reconoce un montón de tipos distintos, pero no el de los archivos con tipo `image/webp`.

Los tipos mime de imagen que WordPress reconoce son:

```
    'jpg|jpeg|jpe'                 => 'image/jpeg',
    'gif'                          => 'image/gif',
    'png'                          => 'image/png',
    'bmp'                          => 'image/bmp',
    'tif|tiff'                     => 'image/tiff',
    'ico'                          => 'image/x-icon',
```

El problema a solucionar, entonces, es que WordPress no reconoce el tipo MIME y queremos que lo reconozca.

Como decía antes, la forma fácil es instalar un plugin, para lo que nos dirigimos al menú `Plugins -> Añadir nuevo` y en la caja de búsqueda, tecleamos algo como «mime type».



Ahora, si queremos ser un poco más atrevidos, tenemos aún dos soluciones posibles:

Solución nº 1
Como WordPress no tiene este `tipo MIME` en la matriz de tipos posibles, considera que es un tipo de archivo peligroso y, es por eso, que no permite que se suba. La solución rápida pero «**NO recomendada**» es decirle a WordPress que ignore la seguridad, lo que hacemos fácilmente introduciendo una línea más en el archivo `wp-config.php`:

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

Solución nº 2
La forma más correcta de hacer el cambio, consiste en cambiar la matriz de tipos reconocidos por WordPress, e insertar el tipo correspondiente entre los de imágenes.
Para esto recurrimos una vez más a nuestro tema hijo, concretamente a modificar el archivo `functions.php` e insertar una función que incluya el tipo que necesitamos en la matriz de tipos aceptados. El código a añadir es:

```
function agrega_mime_type ( $mime_types ) {
 $mime_types['webp'] = 'image/webp';
 return $mime_types;}

add_filter('upload_mimes', 'agrega_mime_type', 1, 1);
```

Si quisiéramos además del tipo «webp» añadir los archivos de tipo «svg» (imagen de tipo vectorial) no tenemos más que agregar otra línea a la misma función, para incluir el tipo que queremos; en este caso: «`$mime_types['svg'] = 'image/svg+xml';`», así que el código final es:

```
function agrega_mime_type ( $mime_types ) {
 $mime_types['webp'] = 'image/webp';
 $mime_types['svg'] = 'image/svg+xml';
 return $mime_types;}

add_filter('upload_mimes', 'agrega_mime_type', 1, 1);
```

WordPress ¿instalo un Plugin o programo?

Es una pregunta recurrente, indudablemente no tiene una única respuesta, dependerá de muchos factores, pero como me preguntan a mí, yo doy mi respuesta.

También hay que tener en cuenta que estamos hablando dentro del mundo de WordPress. Así que

¿Instalo un Plugin o programo?

Indudablemente, si nos referimos a facilidad, en la mayoría de los casos es más rápido y efectivo instalar un plugin.
El mayor inconveniente es, en este caso, escoger correctamente dada la gran variedad de plugins que existen.

Esto puede suponer una gran ventaja y, al mismo tiempo un gran inconveniente, ya que hay que probar varios (instalar, activar, probar, desactivar) para encontrar el que realmente cumple nuestras necesidades.
Luego está el tema de la identificación del problema y la identificación de la mejor solución.

Hay que tener en cuenta que si tu propósito al usar WordPress es simplemente escribir para compartir, es decir, no te interesa en absoluto cómo funciona, lo más aconsejable para ti es instalar un plugin.

Incluso puede darse el caso (bastante común) de que tengas una pareja, un amigo, un conocido, que se encarga del «mantenimiento» por lo que tu sueltas la responsabilidad, le cuentas tu problema o necesidad y … «que se apañe»

Pero es casi seguro que, si estás leyendo esto, no es ese tu caso. Es más, me atrevería a decir que estará de acuerdo con que la respuesta es, sencillamente: **programar**.

Veamos, si instalas un plugin:

- No aprendes
- La mayoría de los plugin resuelven los problemas genéricos, no el tuyo específico.
- La acumulación de plugin no es recomendable.
- Al resolver problemas genéricos, aunque resuelvan tu problema, tienen un montón de código innecesario

Cuidado, no estoy diciendo que no necesites instalar plugins, muchas veces nos facilitan la tarea, solo digo que es conveniente pensar si vale la pena.

Pero al final, sea cual sea tu decisión, ¡disfruta de la publicación! ¡Ah! ¡Felices fiestas!

Un SEO básico

Como con todas las configuraciones de WordPress, si no queremos preocuparnos de hacer un trabajo, tenemos la posibilidad de buscar un plugin que realice esa tarea.

Es más que probable que alguien antes se encontrase con la misma situación y ponga a nuestra disposición un plugin que realiza su cometido.

Un SEO básico.
La cuestión es que se habla mucho del `SEO` y, es bastante enredado, con muchos condicionantes, muchas cosas a tener en cuenta.

Porque además, no existe una ciencia exacta del `SEO`, es algo cambiante. Así que, lo más fácil es hacer caso de las indicaciones de un plugin que nos guíe y se haga cargo de lo necesario.

Hay muchos ejemplos de plugin para `SEO`, no hay más que ir al menú `Plugins -> Añadir nuevo` y buscar la palabra «`SEO`» o, si quieres, en tu navegador vas a la URL https://es.wordpress.org/plugins/search/SEO/, aparecerán muchos dónde escoger.

Pero vamos a ser atrevidos y crear un plugin simple que nos ayude con el `SEO` más básico.

La idea es asegurarnos de tener dos o tres palabras clave por página, que corresponden a esa página y cambiar la descripción para que coincida más con el artículo.

Para ese cometido, usaremos las etiquetas de la entrada actual.

Existe una función dentro de WordPress, que nos hará ese trabajo muy simple: `wp_get_post_tags()`.

Esta función nos devolverá un arreglo o matriz con las etiquetas de la publicación actual. A continuación, daremos forma a esa matriz convirtiéndola en cadena de caracteres y la colocaremos en el encabezado de la página (usando la función `wp_head()`).

Empezaremos creando una función llamada «etiqueta_a_clave»; siguiendo las buenas prácticas de programación, comentaremos el código.

El código de la función es:

```
// SEO
// añadir etiquetas a claves
function etiqueta_a_clave(){
    global $post; // recuperamos la variable global (es un objeto)
    if(is_single() || is_page()){ // verificamos que estamos en una entrada (post) o una página
        $tags = wp_get_post_tags($post->ID); // recuperamos las etiquetas correspondientes
    }
}
```

Ya que `$tags` es un arreglo, podemos comprobar cada uno de sus valores con un bucle `foreach`. A continuación, el código que nos queda es:

```
function etiqueta_a_clave(){
    global $post; // recuperamos la variable global (es un objeto)
    if(is_single() || is_page()){ // verificamos que estamos en una entrada (post) o una página
        $tags = wp_get_post_tags($post->ID); // recuperamos las etiquetas correspondientes
        foreach($tags as $tag){ // recorremos la matriz
            $tag_array[] = $tag->name;
        }
        $tag_string = implode(', ',$tag_array); // implosionamos el contenido
        if($tag_string !== ''){ // Si el contenido NO es nulo
            echo "<meta name='keywords' content='".$tag_string."' />\r\n" // creamos la etiqueta HTML;
        }
    }
}

add_action('wp_head','etiqueta_a_clave');
```

Presta atención a la creación de la etiqueta `html`, podemos añadir (como en este caso) un retorno de carro y un salto de línea (lo que es compatible con sistemas Windows) o solamente un salto de línea (que es compatible con los servidores Linux).

Nos queda entonces, hacer que esta función realice su trabajo, lo que se hace en la última línea de código, llamando a la función de añadir acción (`add_action('action','function')`) pasándole los parámetros «`wp-head`» y «`etiqueta_a_clave`».

Otra de las pautas que tiene en cuenta un plugin de `SEO` es la de publicar como metadato de la entrada, la descripción que tengamos escrita. La mayoría de esos plugin, permiten configurar como metadato de descripción, un texto distinto pero, para hacerlo sencillo, usaremos el establecido.

El código de la función es sencillo y lo adjunto sin comentarios, para que haga el ejercicio de entenderlo (recuerda que si tienes comentarios o preguntas, lo puedes hacer usando el formulario al final de esta entrada).

Recuerda que puedes cambiar la longitud del extracto. ☺

```
// añadir el «excerpt» (extracto) a la descripción
function extracto_a_descripcion(){
    global $post;
    if(is_single() || is_page()){
        $all_post_content = wp_get_single_post($post->ID);
        $excerpt = substr($all_post_content->post_content, 0, 100).' [...]';
        echo "<meta name='description' content="" . $excerpt . "" />\r\n";
    }
    else{
        echo "<meta name='description' content="" . get_bloginfo('description') . "" />\r\n";
    }
}
add_action('wp_head','extracto_a_descripcion');
```

«Breadcrumbs» o «Migas de pan» en WordPress

Las migas de pan son un tema importante no sólo para los humanos quienes visitan tu sitio, sino también para las arañas de los buscadores.

Ah! ¿Como las Cookies?

¡No! No tienen nada que ver. Las Cookies son pequeños archivos con datos que el programa aloja en el ordenador del usuario (como digo en el artículo sobre Cookies). Por otro lado, las migas de pan son indicadores que aparecen en las páginas, para saber el camino de vuelta.

¿Recuerdas el cuento de «Hanssel y Gretel»?

En el cuento, los hermanos usaron migas de pan para marcar el recorrido, de forma que luego sabían, siguiendo el rastro de las migas, el camino de vuelta a casa.

En nuestro caso, las migas también señalan el camino de vuelta, pero a la página de inicio del sitio. De esta forma, sabemos que camino hay que seguir, tanto para volver al inicio como para repetir el camino a la página que estamos viendo.

Veamos un ejemplo, este artículo está alojado en la página de «solowordpress.es», dentro de la categoría «manuales», por lo que podemos volver al principio (Inicio) y podemos volver a este artículo, sabiendo que las «Migas» presentadas son: «`Inicio > Manuales > Breadcrumbs o Migas de pan en WordPress`».

«Breadcrumbs» o «Migas de pan» en WordPress

WordPress no es una excepción, y se considera una buena práctica el uso de las «Migas», el problema es que este tema no se trata de forma nativa en WordPress, así que depende de cada tema, el que puedas poner o no, las «Migas de pan» como parte del contenido.

Como es normal, tenemos solución a ese problema.

La más fácil y rápida, como siempre, es instalar un Plugin que nos haga el trabajo, un ejemplo de ellos es: «Breadcrumb NavXT» (https://es.wordpress.org/plugins/breadcrumb-navxt/) uno de los que más recomendaciones tiene, o también: «Breadcrumb» (https://es.wordpress.org/plugins/breadcrumb/).

Mientras que el primero crea una nueva entrada en el menú, con sus correspondientes enlaces para configuración y uso, este último sólo necesita que incluya un atajo: «[breadcrumb]» en el sitio que deses incluir las «migas de pan».

Pero antes de lanzarte a instalar cualquiera de los plugin para esta tarea, te recomiendo que leas bien la documentación de tu tema, la gran mayoría de los temas modernos ya tienen prevista esta petición y generan ellos mismos las «Breadcrumbs».

Otros plugin no dedicados a la creación de de «Migas», ofrecen la posibilidad de crearlas, como es el caso de muchos plugin de «SEO», ya que como decía antes, también es importante para el posicionamiento.

WordPress y las Cookies

Las Cookies son, nos guste o no, un elemento esencial en cualquier página web interactiva.

WordPress es una página muy interactiva, está continuamente mostrando a los visitantes, lo que han solicitado. Aún en el caso de que «solo» hiciera eso, ha de servirse de algún mecanismo para asegurar que podamos escribir la bitácora y que el visitante puede leerla.

Claro, puedes decirme que para eso ya están los controles de los servidores, del protocolo de comunicación `http(s)`, pero no, el control no debe depender de agentes externos, sino de la aplicación.

WordPress y las Cookies
WordPress es una aplicación desarrollada en `php`, aunque esto sea solo un dato informativo.

Cualquier lenguaje de programación facilita al desarrollador la forma de establecer Cookies.

Y no estaría dando tantas vueltas al asunto de las `Cookies`, si no fuese por la necesidad de cumplir con lo mandado por la ley, concretamente con lo dispuesto en el artículo 22.2 de la Ley 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y de Comercio Electrónico.

Las Cookies son trozos de texto (legible o no por los humanos) que se guardan en el ordenador o dispositivo electrónico del visitante de nuestra web.

Ya que las Cookies son trozos de texto (legible o no por los humanos) que se guardan en el ordenador o dispositivo electrónico del visitante de nuestra web, tenemos el deber de informar a este acerca del uso que vamos a hacer de ellas, ya que necesitamos su consentimiento legal.

Como son trozos de texto y no son código, no es posible usar las Cookies para transmitir ningún programa, sea este benigno o maligno. Tampoco son «Migas de pan» para trazar el camino.

En WordPress las Cookies pueden usarse para muchísimos propósitos, siendo los más usuales el control de:
- La sesión de un usuario registrado.
- Las páginas visualizadas por el visitante.
- Los pedidos de la tienda.
- Las respuestas de una encuesta.

A no ser que seas desarrollador, no tendrás la facultad de establecer tus propias Cookies, pero eso no implica, que no estés obligado a cumplir con el requerimiento legal de avisar a tus visitantes de que tu sitio web usa Cookies.

Puedes ver cómo usamos en **SoloWordpress** las Cookies, visitando nuestra página informativa.
Para cumplir con ese requerimiento, existen una gran variedad de Plugins que te permiten realizar de forma cómoda el aviso a tus usuarios.
En cumplimiento de la ley, deberíamos avisar de forma exhaustiva sobre todas las Cookies que usamos y sobre cuál es su propósito. Como esto no es posible en muchas ocasiones, se hace un aviso genérico, algo como:

Este sitio usa Cookies, si sigues navegando, aceptas su uso.

No es mi cometido dar opiniones legales, así que me limitaré a sugerir que se estudie lo más cuidadosamente posible el plugin elegido, para que cumpla con lo deseado.
Hay muchos plugin, desde los que simplemente informan «de forma genérica», hasta los que pueden configurarse para realizar un informe completo y pueden configurarse con los colores y la forma de aviso que queramos.
Te sugiero que visites la página de plugins y revises las diferentes opciones, antes de instalar un plugin en tu WordPress.

Puedes visitar la página «https://es.wordpress.org/plugins/search/cookies/», elegir y luego, instalar el plugin tras descargarlo o seleccionándolo en la caja de búsqueda en el menú `Plugins -> Añadir nuevo`.

Los «Page Builder» en WordPress 5.3.2

Lo primero es dejar los anglicismos, un «Page Builder» es un «Maquetador Visual» (Aunque no sé si esto lo lía aún más ☺).

Se trata simplemente de un tipo de software que ayuda a diseñar páginas web fácilmente pues, en lugar de tener que editar archivos HTML o PHP, puedes agregar o mover elementos en la página, todo dentro de una interfaz visual. Es por eso que a menudo se les llama creadores de páginas web de arrastrar y soltar (drag'n'drop).

Los «Page Builder» en WordPress 5.3.2
En WordPress casi cualquier tema, permite seleccionar la aparición de uno o varios «widget» en la página y, seleccionar cosas como la posición del logotipo, el formato de la cabecera de página o el formato del pie de página (footer).

Con la aparición de la versión 5, apareció el editor de bloques, que permite redistribuir fácilmente el texto, las imágenes y, otros bloques especiales, en la página o entrada.

Sin embargo, los plugin que te presento, te permiten rediseñar el tema o, mejor dicho, crear tu propio tema, para cualquier página, no solo las entradas de la bitácora.

Estos que te presento, no son todos, pero si son algunos de los más recomendados y están probados con la última versión, la 5.3.2

Page Builder by SiteOrigin
Este es un plugin de maquetación ligero y sencillo. Sin embargo, es uno de los más destacados ya que actualmente cuenta con más de un millón de usuarios activos.

Ofrece una versión gratuita y una Premium, la versión gratuita tiene 23 widgets (incluido el carrusel de posts, slider de imagen y video, y tabla de precios) y 25 plantillas. Puede que a algunas personas les resulte limitada la selección de widgets que ofrece el plugin de SiteOrigin, sin embargo, es compatible con la mayoría de los widgets de WordPress.

Para mi, hay una declaración de intenciones que me resulta atractiva: «Page Builder es nuestro compromiso con la democratización de la creación de contenidos. Como WordPress, Page Builder es, y será, siempre libre. Seguiremos apoyándolo y desarrollándolo tantos años como sea posible. A partir de aquí solo queda mejorarlo.»

Puedes instalarlo a través del menú `Plugins -> Añadir nuevo` y seleccionando en la caja de búsqueda: «Page Builder by SiteOrigin» o, descargarlo desde https://es.wordpress.org/plugins/siteorigin-panels/.

Elementor Page Builder
Este plugin es, según muchos comentarios, el más avanzado (también quizá uno de los más pesados) y a pesar de ser joven (nació en el 2016) tiene ya más de un millón de instalaciones activas.

Elementor para WordPress tiene una versión gratuita, que no tiene tantas funciones como la Premium. Hay tres planes Premium para elegir: Personal, Business y Unlimited.

El plan Personal solo acepta un sitio web. Si estás empezando, este plan será lo mejor para ti. Para tres sitios, puedes comprar el plan Business. El plan Unlimited, por otro lado, acepta cualquier número de sitios web. Todos los planes son anuales.

Puedes instalarlo a través del menú `Plugins -> Añadir nuevo` y seleccionando en la caja de búsqueda: «Elementor Page Builder» o, descargarlo desde https://es.wordpress.org/plugins/elementor/.

WordPress Page Builder – Beaver Builder
Beaver Builder es un maquetador bastante flexible. Resulta bastante ágil y fácil de manejar.

Su punto interesante, es que es uno de los que mejor traducción tiene al español (de España).

Cuenta también con una versión gratuita y una de pago y, como ellos mismos avisan, el soporte para la versión gratuita no es demasiado bueno (queda avisado) así que recomiendan comprar una licencia.

Hay, sin embargo, una creciente colonia, que es como llaman al grupo de usuarios o «castores», a los que puedes dirigirte para pedir ayuda.

Puedes instalarlo a través del menú `Plugins -> Añadir nuevo` y seleccionando en la caja de búsqueda: «Beaver Builder» o, descargarlo desde https://es.wordpress.org/plugins/beaver-builder-lite-version/.

Kadence Blocks – Gutenberg Page Builder Toolkit
Kadence Blocks es uno de los más técnicos, ya que requiere un poco más de conocimiento en la materia de creación de páginas.

Tiene además un pequeño inconveniente, sólo está disponible en inglés, italiano y sueco.

Es por otro lado, bastante potente sobre todo en la creación de contenidos mixtos, como una tabla o matriz con diferentes contenidos (imágenes texto, iframes, etc.)

Puedes instalarlo a través del menú `Plugins -> Añadir nuevo` y seleccionando en la caja de búsqueda: «Kadence Blocks» o, descargarlo desde https://es.wordpress.org/plugins/kadence-blocks/.

Nimble Page Builder
Este modesto plugin, es ligero, está disponible en español de España y es fácil de usar.

Una de las principales razones de su poco peso, es que hace uso de los ganchos y funciones nativas de WordPress, lo que hace al tiempo, que sea compatible con cualquier tema que tengas instalado.

Permite también la creación de páginas desde cero, con lo que no hay porqué seguir las normas marcadas por el tema principal y puedes crear páginas como la famosa 404 o las «legales» sin que necesiten tener la apariencia del resto del sitio.

Puedes instalarlo a través del menú `Plugins -> Añadir nuevo` y seleccionando en la caja de búsqueda: «Nimble Page Builder» o, descargarlo desde https://es.wordpress.org/plugins/nimble-builder/.

Resumen
Ahora que estás listo para crear un sitio de WordPress con un page builder, ten en cuenta que la mayoría de ellos, como los de la lista anterior, ofrecen una función de editor en vivo, al tiempo que ofrecen diferentes paquetes de widgets con precios variables. Por eso debes elegir el que se ajuste a tus necesidades.

Qué es un plugin de WordPress y para qué sirve.

Tanto si acabas de llegar al mundo de WordPress como si empiezas a interesarte por las interioridades, habrás oído mencionar la palabra Plugin (pronunciado «pluguín»).

Qué es un plugin
Un plugin es simplemente un archivo de código, que se añade al código de la aplicación para realizar una tarea no existente en la aplicación base.

Qué es un plugin de WordPress
Este código adicional permite que WordPress tenga una funcionalidad adicional o un comportamiento distinto al que tendría con el código inicial.

Si no existieran los plugin, cada vez que un usuario de WordPress quisiera realizar una tarea que no está dentro de las tareas especificadas, tendría que modificar personalmente el código de WordPress, lo cual está sólo al alcance de unos pocos.

Sin embargo, con los plugin, cualquier usuario puede añadir el código necesario para realizar esa tarea extra, sin necesidad de saber de programación de ordenadores.

Qué plugins existen
Existe una gran variedad de plugins que cumplen una gran variedad de funciones, tantos, que para una misma función es fácil encontrar cientos de versiones.

Por ejemplo, si tu bitácora es sobre fotografía, te interesará buscar un plugin que te permita exponer tus fotos de forma fácil y bonita; sólo has de buscar plugins de fotografía.

Si escribes sobre matemáticas, encontrarás plugins que te facilitan la labor de escribir expresiones y fórmulas.

Si lo que quieres es vender tus productos, existen plugins que facilitan la construcción de una «tienda electrónica».

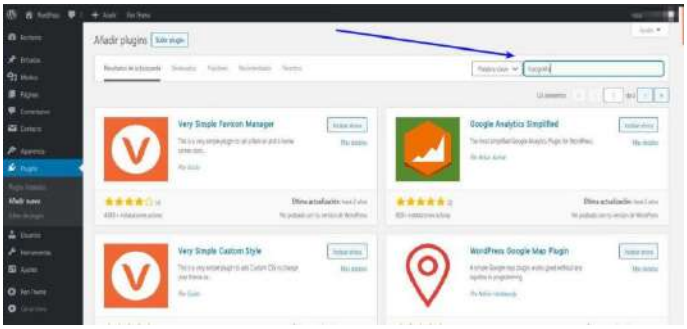
En definitiva, hay de todo, como en botica.

Cómo instalo un plugin
Hay dos formas fáciles y rápidas de instalar un plugin:

Dirígete al menú `Plugins -> Añadir nuevo` (aquí tienes una descripción completa del menú de plugins).

Cuando te aparezca la lista de plugins disponibles para instalar, puedes recorrer la lista completa (ármate de paciencia, hay miles) o usar el buscador, que es la caja que aparece en la parte derecha superior de la lista.

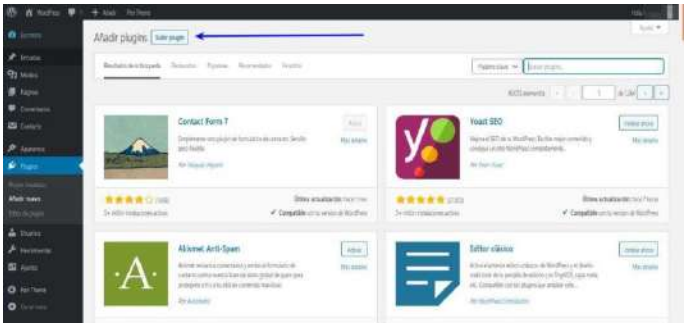
En el buscador entra el término de la clase de plugin que quieres instalar, por ejemplo: «fotografía».



Si bien es casi seguro que aparecerán varios resultados, es decir, varios plugin que cumplen la condición (en este caso, que están relacionados con «fotografía») yo te recomiendo que si no encuentras lo que buscas, pruebes a poner el término de búsqueda en inglés (con la ayuda de un diccionario si es necesario).

Una vez que has encontrado el plugin que buscas, puedes instalarlo directamente o descargarlo. Esto nos lleva al segundo método de instalación.

La segunda forma es instalar el plugin desde un archivo que ya está en tu dispositivo; bien porque lo has encontrado y descargado siguiendo el método anterior, o bien porque te lo «han pasado».



Para esto, dirígete al menú `Plugins -> Añadir nuevo` y, en vez de buscar, pulsa el botón que hay en la parte superior de la página, con la leyenda: «`Subir plugin`», con lo que aparecerá un campo que te permitirá subir el archivo desde tu dispositivo.

Si no encuentras lo que quieres
En el poco probable caso de que no encuentres un plugin que satisfaga tus necesidades, siempre puedes hacer tu mismo un plugin, así conseguirás exactamente lo que quieres.

No es tan difícil como parece, y tienes una forma fácil de comenzar en: Cómo hacer un plugin en WordPress

Naturalmente, esto requerirá que tengas conocimiento de programación en `php` y posiblemente de codificación `html, javascript y CSS`, así que ¿Si quieres lanzarte? ¡Buena suerte!

El error: «Lo siento, no tienes permisos para acceder a esta página» y cómo arreglarlo.

Pocas cosas hay tan frustrantes como intentar acceder al «backoffice» de tu WordPress y encontrarte con una página en blanco y el mensaje: «Lo siento, no tienes permisos para acceder a esta página».

Además de evitar que hagas los importantes cambios en tu sitio, como publicar nuevas entradas, también puede echar por la borda tu arduo trabajo, si no puedes acceder a él.

El error: «Lo siento, no tienes permisos para acceder a esta página» y cómo arreglarlo.
No te alarmes, te voy a contar las razones más comunes para que aparezca este error y, por supuesto, cómo puedes arreglarlo.

Pero antes de entrar en materia, vamos a ver qué genera el error «Lo siento, no tienes permisos para acceder a esta página».

Por suerte, este no es un error muy usual, pero si puedes encontrartelo de vez en cuando, sobre todo si «trasteas» con el servidor donde alojas tu WordPress.

La mayoría de las ocasiones, este error se genera por un problema de seguridad o de permisos. Estos problemas generalmente ocurren cuando cambias de servidor o mueves tu copia de WordPress, que pueden generar discrepancias entre la información que hay en la base de datos y el contenido del disco.

Si los datos almacenados para un tema, plugin o componente del núcleo de WordPress no se corresponden con la versión en la base de datos, las solicitudes no podrán procesarse correctamente y aparecerá el mensaje de error.

Si el nombre de usuario o la contraseña en tu archivo `wp-config.php` no coinciden con la base de datos, aparecerá el mensaje de error; finalmente otra posible causa es si tu sitio está usando una versión obsoleta de PHP (aunque esto es menos probable que ocurra «de golpe»).

Antes de intentar cualquiera de estas acciones, es muy recomendable que hagas una copia de seguridad de tu sitio de WordPress; por supuesto, como no puedes acceder, deberás hacerla con métodos del servidor. Esto asegurará que puedas restaurar fácilmente tu sitio si llegas a cometer un error al tratar de solucionar este problema.

Volver al estado previo al error
Como cualquier otro inconveniente informático, es útil recordar el último cambio que realizaste en el sitio. Después de todo, ese cambio podría haber causado el error.

Podría haber sido cualquier cosa, desde actualizar a una nueva versión de WordPress, hasta migrar tu sitio desde un entorno local, o hacer cambios importantes en PHP. También querrás considerar el último plugin o tema que instalaste (o actualizaste).

Si sabes con certeza cuál fue el último cambio que realizaste en tu sitio, una solución simple es revertirlo. Esto podría significar eliminar el tema o el plugin que crees que esta causando el problema o usar un plugin como WP Rollback (https://es.wordpress.org/plugins/wp-rollback/) para restaurar una versión anterior.

También puedes restaurar tu sitio de WordPress desde una copia de seguridad si tiene una que haya sido creada antes de que se produjera el error por primera vez.

En caso de que no estés seguro de qué cambio realizado podría haber causado el error “Lo siento, no tienes permisos para acceder a esta página”, te puede resultar útil tener acceso a una lista de causas probables.

Obtén una lista de errores con WP_DEBUG
Para obtener una lista de los errores que ha encontrado `PHP` en tu sitio, has de activar la herramienta `WP_DEBUG`.

Para hacer esto, accede mediante una herramienta de `FTP` o a través de tu panel de control (cPanel, Plesk, Webmin, …) y edita el archivo `wp-config.php` que está en el directorio raíz de tu WordPress.

Busca una línea de código que pone:

```
define( 'WP_DEBUG', false );
```

y cambia el valor a «true»

así, tanto si encuentras la línea como si la introduces manualmente, debe quedar:

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

Tras hacer esto, podrás ver en pantalla, si WordPress ha encontrado algún error y la información pertinente.

Mientras tengas activada la función de depuración, se generarán entradas en el registro de errores, que puedes ver en `wp-contents/debug.log`.

Revisa tu acceso y el registro de errores
Además de verificar los errores de PHP dentro de tus archivos, también querrás verificar el registro de errores de tu servidor. Este proceso puede ayudar a eliminar las conjeturas y reducir las posibles causas del error.

Naturalmente, el proceso para revisar tu registro de errores variará según tu proveedor de alojamiento.

Si no puedes detectar una posible causa de este error a través de `WP_DEBUG` o del registro de errores de tu host, puedes obtener más información utilizando un plugin de seguridad.

Por desgracia hay tantas posibilidades en la forma en que tu servicio de alojamiento puede configurar el registro de error, que no puedo decir exactamente dónde o en qué forma puedes verlo. Si no lo sabes o tienes dudas, contacta con tu administrador.

Obtener notificaciones de seguridad con un plugin.
Siempre existe la posibilidad de que el error que experimentas, sea la manifestación de un ataque de «hackers de sombrero negro». Ningún equipo informático es demasiado pequeño o insignificante para no ser blanco de un ataque.

Si tienes instalado un plugin de seguridad que envíe alertas por correo electrónico cuando se produce una actividad sospechosa (como Wordfence Security [https://es.wordpress.org/plugins/wordfence/]), deberías revisar tu correo y tu carpeta de correo no deseado, como primera medida.

Recuerda que hay una gran variedad de plugins de seguridad a tu disposición, que te avisarán si hay un ataque o intento de intrusión. Puedes escoger el que más te guste (https://es.wordpress.org/plugins/search/seguridad/)

Un plugin de seguridad como WP Security Audit Log (https://es.wordpress.org/plugins/wp-security-audit-log/) te permite hacer un seguimiento de los cambios en el núcleo y las configuraciones de WordPress, las actualizaciones del perfil de usuario, las modificaciones de la base de datos y mucho más. Como tal, puede servir como una herramienta útil para identificar la causa raíz de cualquier error

Verificar los permisos.
Una causa bastante probable para que aparezca este mensaje de error, es una desconfiguración de los permisos de archivos.

Existen varios factores que hay que evaluar antes de decir que los permisos de tus archivos son correctos. Aquí te cuento formulas sencillas para corregir este problema.

Una opción que está al alcance de todos, es la de usar tu cuenta de `FTP` para acceder a tus archivos.

Otra opción, la que yo recomiendo, es acceder a través de una cuenta de `SSH`. Supongo que tu servidor es Linux, así que, vamos allá.

Accede mediante tu usuario y contraseña al servidor:

```
ssh mi_usuario@https://miservidor.com
```

A continuación debes cambiar los permiso de los directorios y de los archivos. Podemos hacer esto con dos líneas en la consola, mediante un comando find. Así:

```
cd /htdocs/
$ sudo find /tu_camino_html/wordpress/ -type d -exec chmod 755 {} \;
$ sudo find /tu_camino_html/wordpress/ -type f -exec chmod 644 {} \;
```

Obviamente, si el servidor no es tuyo o no tienes la contraseña de `su`, deberás usar la opción de acceso con `FTP`.
Abre tu cliente de Protocolo de transferencia de archivos (FTP) e ingresa los detalles de tu servidor en los campos relevantes. Luego, ve a la carpeta de html (normalmente public_html). Dentro de ese directorio, tendrás que resaltar los subdirectorios llamados `wp-admin, wp-content y wp-includes`. Luego, selecciona Permisos de archivos y pon los directorios con permisos: `755`

Recuerda marcar el recuadro Repetir en subdirectorios y Aplicar solo a directorios.

La siguiente operación es seleccionar en `public_html` todos los archivos dentro del directorio. Asegúrate de excluir los tres directorios que ya has modificado. Deberás hacer clic derecho sobre los archivos y seleccionar los Permisos de archivo una vez más.

En la ventana de Cambiar atributos, debes poner el valor numérico `644` y, marcar las casillas de Repetir en subdirectorios y Aplicar solo a archivos.

Esto debería resolver el problema, en caso de que fuese causado por problemas de permisos. Recuerda que si tienes instalado algún plugin de seguridad, posiblemente te avise de que debes hacer cambio en los permisos. Hazle caso.

Desactivar Temas y Plugins.
Como ya mencioné, un tema o un plugin que has instalado recientemente, puede causar el error. Por tanto, una buena estrategia es desactivar todos los plugins y tu tema activo. En caso de que no puedas acceder al panel de administración, tendrás que desactivar tus temas y plugins manualmente a través de FTP.

Comprueba si el error ha desaparecido. Si está todo bien, reactiva tu tema y luego tus plugin, uno a uno, probando entre cada activación para comprobar que no aparece de nuevo el error.

Si aparece de nuevo el error «Lo siento, no tienes permisos para acceder a esta página», desactiva nuevamente el último plugin, encontraste el culpable, borralo inmediatamente.

Sin embargo, si este proceso no funciona, mantén abierto tu cliente de FTP.

Regenera tu archivo .htaccess.
Si ninguna de las soluciones anteriores te ha funcionado, podría tratarse de un problema con el archivo `.htaccess`.

Como he dicho en repetidas ocasiones, este es un archivo crítico, así que ten mucho cuidado con cualquier cambio que realices.

El primer paso y fundamental, es mantener el archivo que tienes ahora mismo. Para eso, accede con tu cliente FTP al directorio htm de tu instalación `public_html`.

Localiza el archivo `.htaccess`, haz clic derecho sobre él, selecciona Cambiar nombre y cambia el nombre a `.htaccess_backup`.

A continuación, descarga ese archivo renombrado para poder tenerlo seguro y poder editarlo cómodamente.

Abre el archivo en tu editor de texto preferido (como TextEdit o Notepad o Geany). Borra el contenido del archivo y guardalo como `.htaccess`.

En el archivo en blanco, pon el código básico para WordPress:

```
# BEGIN WordPress
RewriteEngine On
RewriteBase /
RewriteRule ^index.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
# END WordPress
```

Guarda el archivo y subelo al servidor con tu cliente FTP.

Si el archivo `.htaccess` ha sido el causante del problema, ya no tendrás el error. Recuerda que luego deberás verificar el estado de tu plugin de seguridad.

Comprueba la versión de PHP.
Esta es quizá uno de las causas menos comunes de error, ya que no es tan frecuente que sea necesario actualizar la versión del lenguaje PHP.

Sin embargo, una versión obsoleta de este lenguaje, puede causar el error «Lo siento, no tienes permisos para acceder a esta página». Aunque lo más importante es que una versión obsoleta del lenguaje, puede presentar serios problemas de seguridad y estabilidad en tu WordPress.

Así que vale la pena actualizar la versión de PHP que usas. Recuerda que antes de acometer esta acción, debes asegurarte de contar con una copia de seguridad de tu sito (por si acaso).

Yo te aconsejo que instales una copia en local, y pruebes la nueva versión con menos riesgos.

Esta labor es muy delicada e importante y es probable que no puedas realizarla tu mismo y debas pedir al administrador del alojamiento que la realice.

En cualquier caso, en una buena idea mantener actualizado el PHP según las recomendaciones de la versión de WordPress.

Verifica las tablas de tu base de datos.
Otra razón para recibir el error «Lo siento, no tienes permisos para acceder a esta página», es una discrepancia en la información sobre tu base de datos.

Esta situación suele ocurrir cuando se realiza el desarrollo de la página en un servidor local (servidor dedicado o LAMP) y se migra la versión final al servidor online.

Es posible que el prefijo de las tablas en la base de datos, sea distinto en los dos servidores o que el prefijo asignado en la sentencia `$table_prefix = '';` del archivo `wp-config.php` no se corresponda con lo que existe en el servidor `SQL`

Accede a tu servidor `SQL` mediante el método que tengas asignado (por ejemplo `phpmyadmin`). Dirígete a la base de datos que tengas creada para tu WordPress y fíjate en el nombre de las tablas. Deberían empezar todos con lo que se establece en `$table_prefix = '';`; si no es así, cambia el contenido de esta declaración.

O sea, que si la declaración es: `$table_prefix = 'xzy';`, todas las tablas de tu base de datos deben tener un nombre que comienza con '`xyz_`'

Restaura tu sitio WordPress.
Como dije al principio, antes de tomar cualquier decisión, antes de cualquier actuación, hay que hacer una copia de seguridad.

Como vemos en esta lista, hay muchos factores que pueden acabar con la aparición del error «Lo siento, no tienes permisos para acceder a esta página» y es posible que alguno de los pasos tomados anteriormente haya solucionado el problema.

Pero también es posible que el problema persista, con lo cual una última posibilidad es la de restaurar una copia de seguridad de antes de encontrar el error, ya que la única alternativa, sería la instalación de una copia fresca de WordPress, con lo que habrás perdido todas tus páginas y entradas.

Sea como sea, espero que no te encuentres nunca con este escurridizo error.

Conclusión
Como usuario de WordPress, es crucial entender las posibles causas de los errores comunes y saber qué soluciones probar. Hemos visto aquí, cómo atajar un error que puede presentarse, como todos, por sorpresa y, esta, no es una sorpresa ¡agradable!

La Jerarquía de la plantilla y el tema hijo.

En varias ocasiones ya he dicho que una de las grandes ventajas de WordPress es su modularidad y esto se traduce en que es muy flexible.

Y es muy flexible por cómo está diseñado y construido y, gracias a esa flexibilidad, los usuarios podemos crear los llamados tema hijo.

Pero primero, veamos de qué estamos hablando, ¿Qué es la Jerarquía de las plantillas?

Las páginas de WordPress, ya sea una página fija o una entrada, desde el punto de vista del usuario visitante, no son más que un montón de caracteres organizados de forma que el navegador las entiende y, para eso, han de cumplir con las normas de lenguaje de marcado `html` y el `CSS` que les da forma.

Para lograr que las páginas estén en ese lenguaje, WordPress usa un lenguaje de programación que se ejecuta en el servidor, llamado `php`.

Hasta aquí todo es conocido, no descubro nada nuevo, pero entrando un poco más en detalle, es donde empieza la magia.

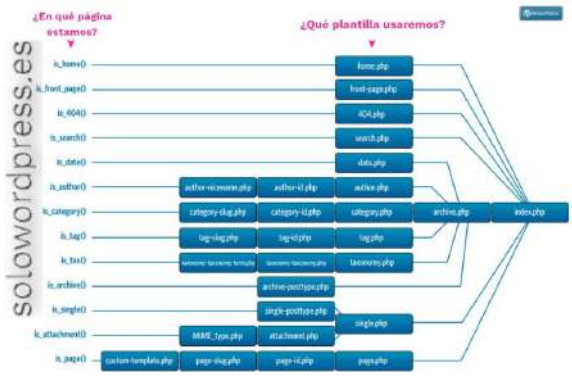
Para que WordPress presente todo en su sitio, como lo hemos dispuesto, usa el lenguaje `php` para tomar las decisiones oportunas y generar el código `html`.

Para que todo sea correcto, WordPress entiende que toda página web, básicamente, tiene tres partes: Una cabecera (`header`) una parte central (`body`) y un pie de página (`footer`).

El `body` a su vez, puede tener o no, espacios laterales `sidebar`, dependiendo de si la distribución está hecha en una, dos, o tres columnas.

Para asegurarse de que todas esas partes están disponibles, hace una serie de preguntas y, basándonos en las respuestas, toma la decisión de usar un archivo u otro.

Quizá esto se ve mejor de forma gráfica. Pero antes de que me acuses de no dejar las cosas claras, te recuerdo que esto es una introducción y no una explicación exhaustiva (que haré en otro artículo).



La idea general detrás de este proceso de decisión, es que la parte central de la página que se quiere representar, contiene uno de los formatos establecidos (es una entrada de una página, es una entrada multipágina, es una página estática, etc.).

Por tanto, hay que construir una página web con cabecera («head»), centro («body») y pie («footer»).

Para hacerlo fácil, vamos a decir que la cabecera y el pie son siempre los mismos, por lo que los archivos `header.php` y `footer.php` del tema que tenemos instalado, serán los que se usen.

Después viene el proceso de decisión sobre qué fichero usamos para la parte central y, es ahí donde se manifiesta la jerarquía de plantillas; se escogerá un archivo u otro, dependiendo de las características de la página que queremos construir.

El proceso de decisión acaba siempre, como última alternativa, el archivo `index.php`, que todo tema debe tener y que, si no hayo otros archivos, puede contener toda la «maquetación» de las páginas.

La Jerarquía de la plantilla y el tema hijo.
¿Qué pasa con el tema hijo?

Dijimos en el artículo sobre Cómo crear un tema hijo en WordPress, que lo único que necesitamos es un archivo functions.php, ¿es falso?

Es cierto, lo que ocurre es que la magia creada por las decisiones de la jerarquía de la plantilla, nos permite usar los archivos (incluido el «index.php» del tema padre, cuando estos no existen en el tema hijo.

Es también por esa razón, que dije que al trabajar con un tema hijo, si hacemos cambios en un archivo del tema padre, debemos situarlo en la misma estructura de directorios que tenga el tema padre, para que pueda encontrarse.

Mi primer plugin de WordPress

Es posible que cuando estás empezando algo, en este caso a sumergirte en el mundo WordPress, te preguntes cómo empezar una tarea y, también es posible, que busques y rebusques información sobre el tema.

Seguro que encuentras miles de artículos que te dicen cosas como, «¡Cómo hacer …», o «la forma fácil de hacer …»

Y cuando empiezas a leer, hay un montón de palabras técnicas y asumen que ya sabes de qué estan hablando. Yo quiero contarte cómo hacer las cosas fáciles y entendiendo lo que haces.

Mi primer plugin de WordPress
Para poder entender las cosas (al menos a mi me pasa) lo mejor es ir por partes, entendiendo cada paso, antes de dar el siguiente.

Lo primero es saber qué es un plugin y para qué sirve.

Puedes leer el artículo Cómo hacer un plugin en WordPress (explicado de forma más general y rápida) o puedes seguir aquí.

¿Qué es un plugin?
Salvando las distancias, un plugin en WordPress es como una aplicación para un sistema operativo. Ambos son trozos de código, en este caso en lenguaje `php`, que realizan una labor dentro del entorno del sistema que los contiene.

En ambos casos, para que sea posible «instalar» ese código dentro del sistema, se han de cumplir unas reglas y requisitos.

Los requisitos para un `plugin` en WordPress son de fácil cumplimiento, pero conviene conocerlos para que las cosas estén correctamente hechas y podamos aprovechar todas las ventajas que el sistema del entorno nos ofrece.

Como es realmente sencillo hacer un plugin, hay plugin hechos para casi todo, la calidad de la programación interna … «Hay de todo, como en botica» y la certeza de que cumplan completamente con nuestros requisitos …

Por eso aplaudo tu inquieteud, nadie hará el plugin que tu necesitas, mejor que tu mismo.

¿Cómo sabe WordPress qué plugin usar?
La respuesta es muy sencilla: No lo sabe.

Jeje, No lo sabe ni puede saberlo. Como he dicho el plugin es un trozo de código que cumple una función. Así, si queremos que WordPress pinte la pantalla de rojo, harémos un plugin que le diga a WordPress que pinte la pantalla de rojo, pero WordPress, no tiene forma de saber que queremos la pantalla de color rojo.

WordPress se limita a recorrer el contenido del subdirectorio `plugins` bajo `wp-content` y ejecutar los trozos de código que encuentre y sean plugin.

WordPress, en su recorrido, no pregunta ni puede saber, qué hace el código, se limita a ejecutarlo

¿Cualquier código?
Jeje, bueno, «Cualquier código» es un poco exagerado, puntualicemos un poco: «Cualquier código que cumpla las condiciones de ser un plugin».

Si situamos un archivo de código `php` que, digamos, realice una suma de los primeros cien números primos, WordPress no lo leerá; pero no porque no sepa sumar, es que no sabe qué hacer con ese código.

Para que un archivo conteniendo un trozo de código en `php` sea considerado un `plugin` debe cumplir una serie de requisitos. Muy fáciles de cumplir, por otro lado.

Los requisitos de un plugin
El archivo que contiene el código, puede tener cualquier nombre. Si hemos podido almacenar el archivo con un nombre y la extensión «.php», el nombre ha cumplido con los requisitos de nombre del servidor y, por tanto, es legible por WordPress (que también está hecho en `php`).

Debe tener un comentario al principio del archivo, a modo de declaración.

Debe contener un código `php` válido.

Si alguna de estas condiciones no se cumple, el `plugin` será ignorado. Después de la versión 5.2 de WordPress, se ha realizado una gran labor para dar más rigidez o, si prefieres al anglicismo, más «resiliencia» al corazón de la aplicación.

Si un plugin está mal configurado, su código no es bueno o, ha quedado obsoleto. WordPress lo desactiva y te avisará con un mensaje de error, para que actualices o corrijas antes de activarlo de nuevo.

Para este ejemplo, vamos a crear un plugin «inofensivo», es decir, que no pasa nada si cometemos un error en su construcción, no podemos romper nada del sistema; ni pasa nada si el plugin no está, ni pasa nada si funciona, no hay nada crítico afectado.

¿Qué hará nuestro plugin?
Para que sea absolutamente transparente para nuestros visitantes, vamos a hacer cambios en un sitio que sólo tu ves, la pantalla de entrada al escritorio de administración o «backoffice».

Así nos aseguramos de que nada de lo que hagamos afectará la buena experiencia de tus lectores.

¡Vamos allá!

Tras instalar WordPress, lo normal es dirigirnos a la dirección https://misitio.xxx/wp-login.php, con lo que aparece la pantalla que nos solicita las credenciales para iniciar sesión. Si no hemos cambiado el aspecto de esa pantalla, veremos algo parecido a:

Para seguir con el ejemlo, suponemos que no hemos instalado ningún plugin de seguridad o de otro tipo que altere el funcionamiento del acceso.

Como habrá más de un acceso a la página por parte de otros administradores, editores, etcétera, queremos personalizar la página y que aparezca el logotipo de nuestra compañía, en lugar del logotipo de WordPress.

Para hacer el cambio, lo que necesitamos es subir el logotipo al servidor, en un sitio accesible a nuestra copia de WordPress. También necesitamos hacer un plugin que haga ese reemplazo.

Vamos a centrarnos en el contenido del archivo del plugin y luego nos ocuparemos de dónde y cómo subirlo al servidor para que nuestro WordPress lo use.

El contenido del archivo
Como el plugin que vamos a hacer modifica la entrada al WordPress, en un alarde de imaginación vamos a llamarle `mi_entrada.php`.

Cómo hemos dicho, la primera parte del archivo es un comentario que resulta identificativo para WordPress. Ese comentario será algo parecid a:

```
<?php
/*
Plugin Name: Mi Entrada a WordPress
Plugin URI: https://misitio.xxx/
Description: Modificación del aspecto de la página de acceso a WordPress
Version: 1.0
Author: [Pon aquí tu nombre]
Author URI: https://misitio.xxx
License: GPLv2 o posterior
*/

?>
```

«Plugin Name» es el nombre que quieres usar para tu plugin; será el nombre por el que aparecerá en la lista de plugins instalados (menú `Plugins -> Plugins instalados`).

«Plugin URI» será la dirección web de la página donde tienes la descripción forma de descarga, etc. de tu plugin (no pasa nada si no la tienes aún).

«Description» Es la descripción de lo que hace tu plugin, la funcionalidad, que aparecerá en la lista de plugins.

«Version» Que corresponde a la versión o revisión de tu plugin.

«Author» Corresponde a tu nombre, como autor del plugin.

«Author URI» Normalmente es la página personal del autor, que no tiene porqué ser la misma que la del plugin.

«License» Por defecto, todo lo que se haga con WordPress está sujeto a la licencia GPL, pero si no es así o tu usas una distinta, ponlo aquí.

La siguiente línea corresponde a la instrucción de «final de comentario» de `php, (*/)`.

La última línea de código es la instrucción de «final de código» de `php, ?>`, esto quiere decir que el código propiamente dicho de nuestro plugin, deberá estar situado entre las líneas de final de comentario (`*/`) y final de código (`?>`).

Lo siguiente que tiene que hacer nuestro plugin, es «hacer algo»; no vale de nada identificarse, si no vamos a realizar ninguna acción.

Pero si dejáramos el archivo como está, habríamos creado un plugin válido para WordPress, aunque no hace absolutamente nada.

Existe mucha documentación sobre WordPress y su funcionamiento, aunque a veces nos encontramos con que no está actualizada y, la actualizada, está en inglés.

En cualquier caso, en esta ocasión, la documentación está actualizada, ya que aunque se han hecho cambios en la pantalla de inicio, especialmente durante las versiones más recientes, lo básico se ha mantenido.

Lo que queremos hacer es reemplazar el logotipo de WordPress por el nuestro y, eso se hace sencillamente, llamando a la acción a un gancho de WordPress.

El código también es muy sencillo y lo explicaremos:

```
// Logo personalizado en login

add_action("login_head", "mi_cabecera_login");
function mi_cabecera_login() {
echo "
<style>
body.login div#login h1 a {
background-image: url('/images/hd-logo.png') !important;
background-size: 155px 101px !important;
background-position: center top !important;
background-repeat: no-repeat !important;
height: 101px;
width: 320px;
}
</style>
";
}
```

Se dice siempre que cada programador tiene su firma, la forma en que se escribe el código indica, muchas veces, quién fue su creador.

Por ejemplo, en este pedacito de código `php`, aprovechando la flexibilidad del lenguaje, esta:

Una línea con el comentario de lo que hace el código que viene a continuación.

Una línea que identifica la acción a realizar y cómo. Voy a explicarla por partes:

```
add_action("login_head", "mi_cabecera_login");
```

«add_action» (es la llamada a la función interna de WordPress que realizará la acción que afectará a una parte del código existente.

«'login_head'» le indica a la función, dónde ha de realizar la acción, en este caso, en la cabecera de la página de entrada («login») en donde debe añadir lo que venga a continuación.

«'mi_cabecera_login'» le dice qué añadir, en este caso, lo que indique el retorno de la función 'mi_cabecera_login'.

A continuación tenemos la declaración de la función

```
function mi_cabecera_login() {
```

Luego viene el cuerpo de la función, es decir, lo que queremos que haga y, en este caso, es una única línea de código, una sentencia `echo`, con todo el contenido que queremos insertar en la cabecera.

La línea `echo` introduce en la cabecera de la página, la declaración de estilo que modifica la existente, para poner la imagen del logotipo.

La forma de alterar la declaración existente, es:

1) Especificando lo más concretamente posible, el elemento sobre el que actuamos: `body.login div#login h1 a`
2) Diciendo al CSS, que es importante el cambio: `!important`

Si analizas el código, te habrás dado cuenta de que podríamos haber resumido todo esto en una sola sentencia, en lugar de usar una función con un echo, es decir, que retorne una cadena de caracteres, podríamos introducirla directamente. Algo como:

```
// Logo personalizado en login

add_action("login_head", "<style>
body.login div#login h1 a {");
```

Y todo lo que viene a continuación, pero hay varias razones para no hacerlo, entre ellas:

- **Segmentación**. Si ocurre algún error, es más fácil encontrarlo si tenemos el punto localizado, no sólo una laaaarga cadena de caractéres
- **Claridad**. Aunque el motor de PHP sea capaz de entender esa larga ristra de caracteres, nosotros simples humanos, tenemos dificultad en manejarla, hacerlo línea a línea es mas cómodo.

Y con esto concluimos la clase de hoy … No, ya puestos, vamos a incluir un par de cambios más.

Si lo dejamos así, el plugin cambiará el logotipo, pero si alguien hace clic (o toca) el logotipo, será redirigido a «http://wordpress.org», en lugar de a nuestro sitio web.

Vamos a cambiar ese comportamiento, para que las cosas concuerden.

WordPress ya tiene previsto el que los usuarios quieran cambiar esto, por lo que existe una función interna llamada «'login_headerurl'» que podemos alterar fácilmente.

Agregamos al código de nuestro plugin:

```
// personalizar url logo acceso
add_action( 'login_headerurl', 'mi_acceso_personal_url' );
function mi_acceso_personal_url() {
return 'https://misitio.xxx';
}
```

Y también podemos cambiar lo que aparece si algún visitante ve la página con un lector, exactamente, cambia el contenido de la «h1» (que actualmente es: «funciona gracias a wordpress»)

El código es:

```
//Cambiar texto alt del logo de login
add_action("login_headertitle","my_custom_login_title");
function my_custom_login_title()
{
return 'Bienvenido a HD';
}
```

Ahora si, tenemos completo nuestro primer plugin de WordPress.

Ahora nos queda situarlo en un sitio adecuado para que realice su función en nuestro WordPress.

Por un lado, tenemos el archivo del código `php` llamado `mi_entrada.php` y por otro, el logotipo que queremos reemplazar, llamado `hd-logo.png`.

Este último, como hemos indicado en el código, deberá situarse en un subdirectorio llamado `images`, debajo del directorio raíz de nuestra instalación de WordPress.

Por su lado, el `plugin` propiamente dicho, deberá estar dentro de una carpeta o subdirectorio situado bajo el subdirectorio `plugins`, con el nombre `mi_entrada`, tendremos así un camino o «path»: `/wp-content/plugins/mi_entrada/mi_entrada.php`

Al acceder al menú `Plugins -> Plugins instalados` aparecerá un plugin con el nombre de «mi_entrada» que estará desactivado. Una vez se active, empezará a cumplir su función y podremos apreciar los cambios en la pagina de acceso.

Las Bondades de Gutenberg

Es innegable que Gutenberg (nombre «en clave» del editor de bloques) ha venido para quedarse, así que mejor aprendemos a sacarle partido en lugar de luchar contra el cambio.

Los que llevamos un tiempo trabajando con WordPress, nos hemos acostumbrado a una forma de trabajo, a tener nuestras herramientas de edición y de maquetación, en una posición determinada, etcétera.

Como ocurre en todas las profesiones, el cambio siempre es más fácil para unos y, menos fácil para otros.

¿Qué es Gutenberg?
Gutenberg es el nombre que se le ha dado al editor de bloques de WordPress que apareció en la versión 5.0

Se trata de un esfuerzo por modernizar la edición en el CMS y que ha presentado un reto a los que trabajamos con esta plataforma, ya que supone un cambio de paradigma.

De igual manera que supuso un cambio de paradigma la nueva forma de impresión que fue el invento de **Johannes Gutenberg** allá por el año 1456. (Aunque el invento sea anterior, fue en ese año cuando publicó «La Biblia» impresa con su invento).

He de puntualizar que, en contra de lo que muchos afirman, **Johannes Gutenberg**, NO inventó la imprenta (que ya usaban los chinos mucho tiempo antes) sino la imprenta de tipos móviles, que agilizó sobre manera todo el trabajo de impresión.

No se si el editor de textos supondrá un cambio tan importante como su nombre sugiere, pero al menos, tenemos claro que es un cambio con proyección de futuro, así que mejor aprender cómo funciona.

Las bondades de Gutenberg
Al igual que con el editor clásico, lo común y quizá más indicado, es escribir párrafo tras párrafo sin ocuparse de lo que se conoce como maquetación (darle estilo).

Una vez terminado el, por otro lado nada fácil, trabajo de escribir, se procede a dar forma al escrito, insertando encabezados donde proceda, poniendo algunas palabras resaltadas en «negrita» o «cursiva» y realizando las correcciones oportunas.
Quizá deban insertarse imágenes esclarecedoras o simplemente embellecedoras.
Mientras que se está escribiendo, cada vez que pulsemos la tecla «Intro» o «Enter» o «Entrar» (según disponga tu teclado), Gutenberg finalizará el bloque de párrafo y empezará un nuevo bloque (también de párrafo) automáticamente.

Si tu forma de escribir así te lo permite, antes de empezar automáticamente con un nuevo bloque de párrafo, tienes la posibilidad de indicar qué clase de bloques vas a realizar a continuación (imagen, encabezado, HTML, lista, …)

Muchos de los que estamos acostumbrados ya a la manera clásica, echamos de menos la cantidad de posibilidades (sobre todo porque seguro que tenemos una buena colección de plugins al respecto) que nos ofrecía el editor TinyMCE
Contando con la barra de herramientas, podíamos fácilmente añadir emoticonos, emoji, cambiar el tipo de letra para una o varias palabras, etcétera.

La barra de herramientas de Gutenberg, es aún bastante limitada, y nos ofrece (en un bloque de texto) las posibilidades de (en la parte central, en los extremos hay dos botones que luego explicaré y que son comunes a todos los tipos de bloque):

- Cambiar la lineación del texto a la izquierda, al centro, a la derecha.
- Cambiar las palabras seleccionadas a «negrita» (B por su nombre en inglés «bold»).
- Cambiar las palabras seleccionadas a «cursiva» (i por su nombre en inglés «italics»).
- Convertir las palabras seleccionadas en un hyper enlace (imagen de una cadena de tres eslabones).
- Un pequeño triangulo apuntando hacia abajo, nos ofrece las posibilidades de
  - Código integrado (inscribe las palabras seleccionadas entre «<code>» y «</code>»).
  - Imagen integrada (inserta una imagen de la galería de medios, en la posición en donde se encuentra el cursor).
  - Tachado (inscribe las palabras seleccionadas entre «<s>» y «</s>», el tachado de html [stroke, in inglés]).

Por su parte, si el bloque que estamos escribiendo es uno de tipo «lista», la barra de herramientas nos ofrece las posibilidades adaptadas a listas, que son:

- Convertir a lista desordenada (si no lo es ya, jeje hace la lista con un html «<ul>»).
- Convertir a lista ordenada (hace la lista con un html «<ol>»).
- Reducir sangría al elemento de la lista (reduce el nivel de indentación del elemento en el que está situado el cursor).

Los siguientes elementos, son iguales a los de la cabecera para el bloque de párrafo.

Los dos extremos de la barra de herramientas que antes dije que eran comunes (no solo a estos dos tipos de bloque sino a todos los tipos) son:

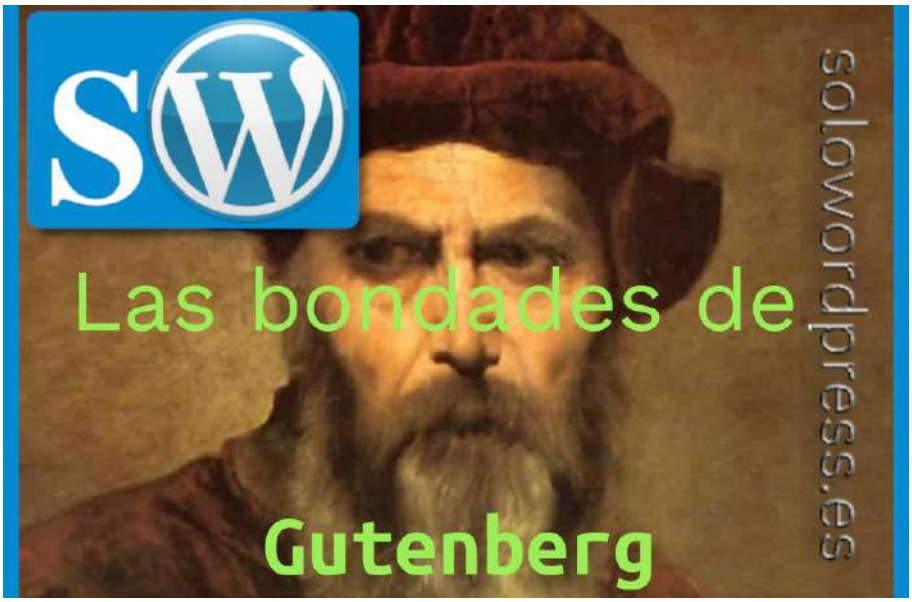
- En el lado izquierdo, un botón que cambia de apariencia en cuanto situamos el cursor sobre él. Cambia de la forma indicativa del bloque en el que estamos, a un par de flechas curvas que, al pulsarlo, nos permite cambiar el tipo del bloque en el que nos encontramos.
- En el lado derecho, el icono de tres puntos en disposición vertical que nos indica que es un menú desplegable y que tiene las siguientes opciones (marcado en gris en el borde derecho, los atajos de teclado correspondientes):
  - **Ocultar los ajustes del bloque.** Que ocultará la barra a la derecha, donde aparecen las opciones relativas a las características del bloque en el que estamos.
  - **Duplicar**. Que duplica debajo, el bloque en el que nos encontramos.
  - **Insertar antes**. Que inserta un bloque justo antes de aquel en el que estamos.
  - **Insertar después**. Que inserta un bloque justo después de aquel en el que estamos, desplazando hacia abajo los demás bloque que haya.
  - **Editar como HTML**. Que transforma el editor del bloque en el que nos encontramos, para presentar el código html (útil si queremos insertar etiquetas con «» o «<abbr>» que no tenemos disponibles en la barra de herramientas.
  - **Añadir a los bloques reutilizables**. Añade el bloque en el que nos encontramos, a la lista de bloques reutilizables. Podremos después, llamar al bloque por su nombre y usarlo (una copia) en otro lugar.
  - **Agrupar**. Podemos marcar el bloque en el que estamos y luego uno o más contiguos, para convertirlos en un solo bloque.
  - **Eliminar el bloque**. Elimina totalmente el bloque y su contenido.

Existen otros muchos tipos de bloques, como el de imagen, el HTML personalizado o el de Galería. Cada uno de ellos cuenta con su peculiar grupo de características básicas editables con la barra de herramientas.

Resumen

Gutenberg ha llegado para quedarse, no está absolutamente todo depurado, como con todas las aplicaciones, irá mejorando en calidad y opciones, con la ayuda de la comunidad y el paso del tiempo.

Aprovecha ahora para ir aprendiendo a usarlo y disfrutarlo, plantea tus dudas y necesidades. Entre todos haremos que esto vaya mejorando. 😀



Alerta de seguridad

Dos plugin para WordPress, **InfiniteWP Client** y **WP Time Capsule**, contienen una seria vulnerabilidad que ha expuesto aproximadamente unos 320.000 sitios.

Este par de plugins, se usan para administrar varios sitios con WordPress desde un único servidor y crear copias de seguridad de archivos y bases de datos cuando aparecen actualizaciones.

El problema
Los dos han sido examinados por la empresa de ciber seguridad **WebArx** y, en su informe, dicen haber encontrado «problemas lógicos en el código que le permite iniciar sesión en una cuenta de administrador sin contraseña».

InfiniteWP está activo en más de 300,00 sitios, mientras que WP Time Capsule está activo en al menos 20,000 dominios, según la librería de WordPress.

Según las declaraciones hechas el pasado martes, el impacto en WP Time Capsule se produce en las versiones anteriores a la 1.9.4.5, en donde es posible usar un requerimiento POST con un componente JSON codificado en Base64, para saltarse los requerimientos de seguridad y acceder sabiendo sólo el nombre de un usuario administrador.

En el caso de WP Time Capsule, las versiones afectadas son los anteriores a la 1.21.16, en las que una línea de una función permite que se cree una cadena de caracteres en un requerimiento POST que llama a otra función que obtiene todas las cuentas de administración y accede con las credenciales del primer administrador de la lista.

La compañía WebArx ha informado sobre las vulnerabilidades en el desarrollo de estos plugin el pasado 7 de enero. Ambos autores de los plugin han reaccionado rápido y han lanzado actualizaciones parcheadas durante las siguientes 24 horas.

Así mismo, los desarrolladores dicen haber editado medidas de seguridad adicionales para evitar las peticiones POST maliciosas.

Recomendamos encarecidamente a nuestros lectores que usan esas herramientas, que actualicen lo antes posible los plugin correspondientes.

La empresa de seguridad, en su comunicado advierte: «Es difícil bloquear esta vulnerabilidad con las reglas generales del firewall porque la carga útil está codificada y una carga maliciosa no se vería muy diferente en comparación con una carga útil de aspecto legítimo de ambos complementos.»

Por su parte, sobre los desarrolladores han manifestado: «Siempre es bueno ver a los desarrolladores que están tomando medidas rápidamente y están informando a sus clientes sobre los problemas para ayudar a las personas a actualizar a una versión más segura lo antes posible.»

Alerta de Seguridad [20200117]

Alerta de Seguridad del 17 de enero de 2020

En una semana «negra», nos encontramos hoy con una nueva alerta.

Hoy la noticia la protagoniza un plugin con un serio problema de seguridad, se trata del WP Database Reset.
Se ha descubierto que este plugin presenta un comportamiento tal que un usuario cualquiera, sin necesidad de ser administrador, puede borrar cualquier tabla que desee de la base de datos.

Además, otra vulnerabilidad descubierta en su comportamiento es que el usuario, tenga el nivel de autorización que tenga, puede elevar sus privilegios a «nivel Dios» y eliminar todas las entradas de la tabla «users», quedando él como único usuario con todos los privilegios.

En el momento de escribir esta noticia, los autores del plugin están sacando la versión corregida, la versión 3.15

Si estáis usando una versión anterior, debéis actualizar lo antes posible, aún si no está disponible en vuestra versión de WordPress.

Internationalization

La Internacionalización en WordPress

Este título tan largo y en inglés… Lo cierto es que está hecho adrede, su traducción al español no es tan difícil ¿o si?.

Vamos a hablar de un término un tanto polémico, no todos le llaman así y muchos discuten su significado pero, en fin, vamos a ver de qué se trata.

¿Qué es la Internacionalización?
La internacionalización es el proceso de desarrollar tu tema o plugin, de forma que sea fácilmente traducible a otras lenguas. A menudo, este término se abrevia como i18n, porque hay 18 letras entre la «i» y la «n»

¿Por qué es importante la internacionalización?
El CMS WordPress se usa en todo el mundo, y si, en muchos países, ni el inglés ni el español, son la lengua principal. Por tanto, debemos codificar los plugin y los temas, en una forma en la que sea fácil traducirlos a otras lenguas. Como desarrollador, puede que no puedas facilitar tus creaciones en todas las lenguas de tus usuarios, pero un traductor puede hacer ese trabajo si solo ha de traducir y no codificar él mismo.

¿Cómo internacionalizar tu tema o plugin?
Para que los textos en el tema o plugin sean traducibles, no deben estar fijos en el sitio, sino ser pasados como argumentos de una función de «localización» de WordPress.

El siguiente ejemplo muestra un código fijo, que no podría ser traducido sin cambiar el archivo que lo contiene, lo cual no es muy eficiente.

```
<h1>Settings Page</h1>
```

Sin embargo, pasando la cadena de caracteres como parámetro de una función de localización, puede ser fácilmente traducida.

```
<h1><?php _e(
'Settings Page'
); ?></h1>
```

WordPress usa las librerías **gettext** para poder añadir las traducciones en PHP. Deberías usar las funciones de localización de WordPress en lugar de las nativas de PHP.

El «Text Domain» o «Dominio del texto»
En las funciones de internalización, como segundo argumento tenemos el «text domain». Se trata de un identificador único que permite a WordPress distinguir entre todas las traducciones que maneja. El «text domain» sólo es necesario definirlo en temas y plugins.

Para los temas que están albergados en WordPress.org, «text domain» debe ser igual al «slug» de la URL del tema o plugin (wordpress.org/themes/<slug>); esto es necesario para que las traducciones funcionen correctamente.

Existen unas reglas sencillas para el «text domain», deben usarse guiones (–) y no «guión bajo» o «subrayado» (_) y debe estar en minúsculas. Por ejemplo, si en la cabecera del archivo style.css del tema se hace a: mi tema, o el tema está almacenado en un directorio, el «text domain» debe ser: «mi-tema».

El «text domain» se usa en tres sitios distintos:

1. En el style.css de cabecera del tema.
2. Como un argumento de las funciones de localización.
3. Como un argumento al cargar las traducciones usando load_theme_textdomain() o load_child_theme_textdomain()span>

La cabecera del style.css del tema
El «text domain» se integra en la cabecera del archivo style.css para que el meta-dato de la descripción pueda ser traducido aún si el tema no está activo.

Ejemplo:

```
/*
* Theme Name: Mi tema
* Author: Autor del tema
* Text Domain: mi-tema
*/
```

Camino del dominio (Domain Path)
El Domain Path se necesita cuando las traducciones están almacenadas en un directorio distinto de languages. De esta forma, WordPress sabe dónde encontrar las traducciones cuando el tema no está activo. Por ejemplo, si los archivos ‘ .mo’ están almacenados en el directorio idiomas, el Domain Path será /idiomas y deberá estar escrito con la barra inclinada incial (/). Por defecto, se asume la carpeta languages en el mismo directorio del tema.

Ejemplo:

```
/*
* Theme Name: Mi tema
* Author: Autor del tema
* Text Domain: mi-tema
* Domain Path: /idiomas
*/
```

Añadir «text domain» a las cadenas (string)
Para que las traducciones funcionen correctamente, el «text domain» deberá estar como argumento en todas las llamadas a las funciones de localización.

Ejemplo Nº 1:

```
__(
'Post'
)
```

Deberá ser

```
__(
'Post', 'mi-tema'
)
```

Ejemplo Nº 2:

```
__e(
'Post'
)
```

Deberá ser

```
__e(
'Post', 'mi-tema'
)
```

Ejemplo Nº 3:

```
__n(
'One post', '%s posts', $count
)
```

Deberá ser

```
__n(
'One post', '%s posts', $count, 'mi-tema'
)
```

Atención: El «text domain» deberá pasarse como una cadena y NO como una variable. Esto permite que las herramientas de análisis diferencien entre distintos «text domain». Por ejemplo, esto NO debe hacerse:

```
__( 'Traduzcanme', $text_domain);
```

Cargando las traducciones en WordPress
Las traducciones en WordPress se almacenan en archivos ‘ .po’ y ‘ .mo’ que deben ser cargados usando las funciones de load_theme_textdomain() o load_child_theme_textdomain(). Esto cargará el archivo {locale}.mo desde el directorio de idiomas del tema de WordPress en /wp-content/languages/themes/(tema)/.

Nota: A partir de la versión 4.6 de WordPress, se comprueba automáticamente el directorio en /wp-content/ en busca traducciones descargadas desde translate.wordpress.org. Esto quiere decir que los plugins que hayan sido traducidos usando «translate.wordpress.org», no necesitan la función `load_plugin_textdomain()`. Así que si no quieres usar la función `load_plugin_textdomain()` tienes que especificar en la descripción de tu plugin (el archivo readme.txt): Requires at least: 4.6.

Si quieres saber más sobre los diferentes lenguajes y sus correspondientes códigos de país, visita la lista en https://make.wordpress.org/polyglots/teams/.

Cuidado:

1. Recuerda renombrar el archivo MO a ‘{locale}.mo’ (P.e.: es_ES.po y es_ES.mo) si pones las traducciones en la carpeta del tema o plugin.
2. Renombra tu archivo MO como ‘{text-domain}-{locale}.mo’ (P.e.: mi-tema-es_ES.po y mi-tema-es_ES.mo) si pones las traducciones en la carpeta de idiomas de WordPress.

Ejemplo:

```
function my_theme_load_theme_textdomain() {
    load_theme_textdomain(
    'mi-tema', get_template_directory() . '/idiomas');
}
add_action( 'after_setup_theme', 'my_theme_load_theme_textdomain');
```

Idealmente, esta función debe ser parte del archivo functions.php del tema.

Paquetes de Idiomas
Si te interesa el tema de los paquetes de idiomas y de cómo importarlos a translate.wordpress.org, hay una buena documentación en inglés en: https://make.wordpress.org/meta/handbook/documentation/translations/.

Internacionalizando tu tema
Ahora ya sabes como hacer que tus creaciones (temas y plugins) estén preparados para ser traducidos a otros idiomas.

Te recomiendo la lectura de un artículo (en inglés) sobre la internacionalización, escrito por los creadores de la API. Encontrarás información y consejos para unas buenas prácticas.
Visita: https://developer.wordpress.org/apis/handbook/internationalization/.

Mi experiencia con Gutenberg

Mi experiencia con Gutenberg

Tengo que contarlo si no, reviento. Aunque puedes decir «Y a mí qué me importa»; por mi salud mental, debo contarte

Mi experiencia con Gutenberg
Durante la última semana, he cumplido un reto personal.

Como sabes, tras la aparición de la versión 5.0 de WordPress, se instaló como «aconsejado» y «por defecto» el editor Gutenberg.

Ese editor supone un gran esfuerzo por parte de los desarrolladores de WordPress, un editor «visual» no es algo trivial.

Mi deber no solo conmigo mismo, sino contigo, es contarte las cosas lo más certeramente posible. Para eso, debo contar acerca de lo que conozco y, no puedo conocer una cosa, si no la pruebo.

Por otro lado, debo cumplir con mi misión de divulgación y decirte que el editor Gutenberg o editor de bloques, es (según «los padres de la criatura») el futuro, y debo recomendar su uso.

Volviendo a mi reto personal, el reto consistió en usar el editor de bloques para crear los artículos que publico en esta bitácora. Eso suponía editar un nuevo artículo cada día, usando el editor de bloques.

La conclusión: «**experimento fallido**».

Reconozco mi incapacidad para manejar ese editor de forma eficiente, incluso un artículo que usualmente me demanda como mucho 1 hora de trabajo, se convirtió en una lucha derivada en 20 horas de trabajo.

Es cierto, no puedo decir que el editor no funcione, pero si puedo afirmar que hoy por hoy, aún hay muchos plugin que no se han adaptado a este nuevo editor y que necesito para mi trabajo, con lo que me he visto forzado a «hacer malabares» para poder completar tareas simples.

Quizá para el escritor que lo único que necesita es escribir texto puro; de vez en cuando un encabezado y quizá una o dos imágenes; quizá, decía, el editor de bloque pueda resultar cómodo y útil.

Si lo que tienes en tu bitácora, implica cambios de estilo en el texto o en la distribución del texto con respecto a otros bloques de texto o imagen, el editor de bloques NO es para ti.

Nos han prometido una mayor agilidad, pero eso aún no ha llegado.

Como ya conté, puedes usar el **bloque clásico** y ahí trabajar de forma parecida a como lo hacías con el editor clásico, pero también limitada.

Mi mensaje a WordPress es: ¡Buen esfuerzo, señores! Aún no está suficientemente maduro para mí, gracias.

Seguiré usando el editor clásico hasta que tenga la oportunidad de ver que el editor de bloques realmente me facilita el trabajo.

Gutenberg deja rastro, aunque no se use.

Gutenberg deja rastro, aunque no se use.

Estos días pasados he estado dando una nueva oportunidad al editor de bloques en WordPress 5.3.2 (Gutenberg, para los amigos). Mi experiencia no ha sido muy alentadora, como expresé en Mi experiencia con Gutenberg.
Pero he seguido investigando dándole vueltas, porque es posible que sea yo el que no se adapta. 😀

Y mi sorpresa ha sido cuando me he encontrado con que Gutenberg deja rastro, aunque no se use.

Gutenberg deja rastro, aunque no se use.
Aunque he desactivado el editor de bloques, he visto que sigue existiendo un rastro en la cabecera de las páginas generadas por WordPress.
Esto no tendría mayor importancia si estamos seguros de que el servidor en el que se encuentra nuestro WordPress, está optimizado.
Es posible que unos mili segundos no atasquen el servidor, pero yo siempre pienso que mientras menos cosas innecesarias, mejor.
Se trata de un archivo de tan sólo 40,49 kB (en la versión 5.3.2 en español [es_ES]), lo que no supone una gran carga, pero …
Así que me dispongo a eliminarlo. El archivo es un CSS minimizado, que se usa en caso de que las entradas o páginas estén creadas con Gutenberg y, por tanto, se debe respetar el estilo.
Para eliminar la llamada a ese archivo en la cabecera (<head></head>) hago una llamada en el archivo functions.php:

```
add_action( 'wp_enqueue_scripts', 'twp_remove_gutenberg_block_css', 100 );
function twp_remove_gutenberg_block_css() {
    wp_dequeue_style( 'wp-block-library' );
}
```

Este simple código, quita de la cola de tareas, incluir en el <head></head> la llamada a la carga de este archivo.

WordPress y los botones transparentes

Lo que ocurre con las modas es que son efímeras, rápidas y, muchas veces, inexplicables.

Se ha puesto de moda o, dicho en moderno, es tendencia, que los enlaces en las páginas se manifiesten como botones, aunque no lo sean.
WordPress no es ajeno a las modas y esto lo digo, porque ya varias personas me han manifestado su deseo de tener esa característica en sus bitácoras. Así que vamos a ver qué es ese invento.

WordPress y los botones transparentes
Cuando escuché la petición la primera vez, me quedé perplejo, no entendía para que se quiere poner un elemento transparente en una página web, así que pedí explicaciones y me contaron de qué estaba hablando.

Los botones transparentes ni son botones, ni son transparentes.
De lo que se trata es de que todos los enlaces se comporten, en parte, como botones.

Se quiere que los enlaces presenten un aspecto de botón cuando el puntero del ratón se sitúa sobre ellos.

La mayor preocupación para las personas que me hacen la pregunta, es que tengan que modificar el tema que están usando, o crear un tema hijo, que en cualquiera de los casos (para ellos) significa contratar a un diseñador para que les haga ese trabajo.

Pues voy a darles una alegría, ¡me siento generoso!

No hay que hacer ninguna de esas cosas (en la mayoría de los casos).

Salvo que estés usando un tema realmente elaborado, la mayoría se limitan a hacer pequeños cambios en el color y la decoración de los enlaces y, aprovechando la potencia de WordPress, vamos a crear nuestros botones transparentes.

Necesitamos solo un poco de paciencia, el trabajo es sencillo y lo haré paso a paso.

Todo consiste en un poco de «magia» CSS.

La mayoría de los temas, nos permiten añadir unas cuantas reglas CSS a lo que tienen establecido y, esto se hace a través del menú Apariencia -> Personalizar -> CSS adicional; ahí pegas el siguiente código base CSS:

```
.boton-transparente {
  display: inline-block;
  padding: 8px;
  color: #eee;
  border: 2px solid #fff;
  text-align: center;
  outline: none;
  text-decoration: none;
  transition: background-color 0.2s ease-out,
              color 0.2s ease-out;
}
.boton-transparente:hover,
.boton-transparente:active {
  background-color: #ccc;
  color: #000;
  transition: background-color 0.3s ease-in,
              color 0.3s ease-in;
}
```

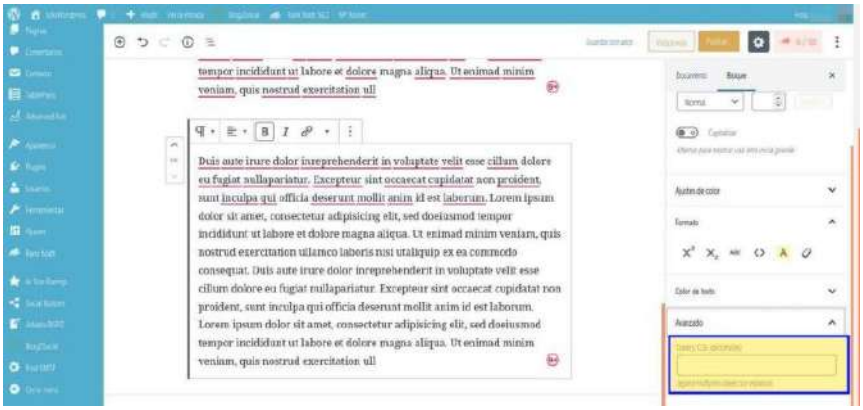
Normalmente no habrá conflictos con otros enlaces, pero por si acaso, no hemos querido cambiar «todos» los enlaces, cambiarás sólo los que desees.

Cómo funciona
Una vez que hemos creado esta sencilla clase CSS, la podemos usar con cualquier elemento.

Si usas el editor clásico o un bloque html, deberás especificar la clase en el elemento, por ejemplo:

```
<a class="boton-transparente" href="https://enlace-del-
boton.com">Texto del botón</a>
```

Si eres de los que usa el editor de bloques, por ahora sólo puedes usar bloques html para especificar lo que quieres hacer, porque los bloques de párrafo o de lista o de cabecera, sólo permiten especificar la clase para todo el bloque.



reCaptcha V3 en el formulario de contacto

Curiosamente, el formulario de contacto, es una de esas cosas a las que no se les suele prestar mucha atención, sin embargo es primordial para muchos.

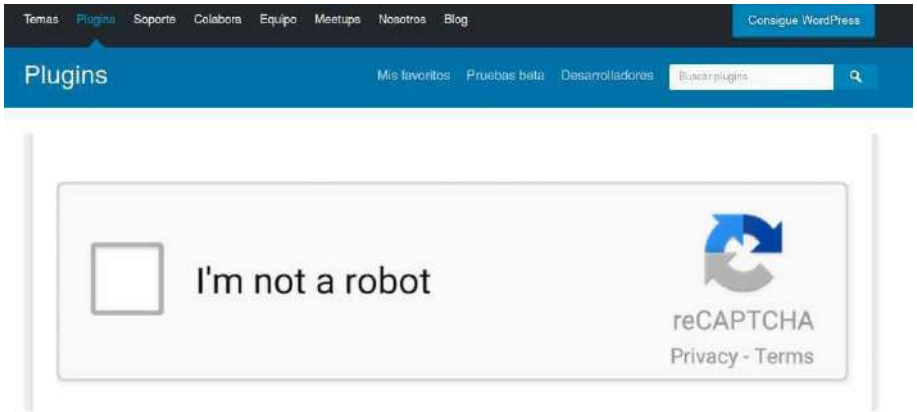
Quizá esto sucede porque suele ser un elemento que no es necesario cambiar continuamente y, por otro lado, es relativamente sencillo.

A esto le sumamos el factor conocimiento, es decir, aunque hay muchos plugin de formularios, casi todos ofrecen las mismas características fundamentales y el más conocido, es el «Contact Form 7» (https://es.wordpress.org/plugins/contact-form-7/).

La ventaja de ser el más conocido (y uno de los más usados) es que muchos desarrolladores trabajan para «mejorarlo» o hacer ampliaciones de sus posibilidades.

Una de las posibilidades no básicas (que necesita ampliación, no viene «de serie») es la posibilidad de integrarlo con un sistema de seguridad.

Sistemas de seguridad o «securización» de formularios también hay varios, pero uno de los más usados es el de reCaptcha de Google (https://www.google.com/recaptcha/intro/v3.html).



reCaptcha V3 en el formulario de contacto
reCaptcha se ha actualizado a la versión V3, lo que ha supuesto un inconveniente para los usuarios de ese sistema de seguridad que lo tengan integrado con Contact Form 7; de repente, el sistema de seguridad ha dejado de funcionar.

Debido a la popularidad de ambos elementos, los desarrolladores de plugin se han apresurado a actualizarse, por lo que es relativamente fácil arreglar el inconveniente, tan solo se tiene que instalar un plugin que ofrezca la integración con el nuevo modelo de reCaptcha, como por ejemplo el Advanced noCaptcha & invisible Captcha (v2 & v3).

También hay quien «se niega a cambiar» y prefieren usar la versión antigua (V2) del sistema de seguridad aunque esto genera otro inconveniente, al actualizar el plugin de formulario, se pierde la compatibilidad con el sistema de seguridad antiguo.

Es decir, Contact Form 7 hasta la versión 5.0.5, funciona con reCaptacha V2; Contact Form 7 V5.1 funciona con reCaptacha V3.

La solución para poder seguir utilizando reCaptacha V2 pasa por instalar un plugin que realice esa «magia», como ReCaptcha v2 for Contact Form 7 (https://es.wordpress.org/plugins/wpcf7-recaptcha/).

El problema de incompatibilidad, parece que reside en el código del plugin de contacto y, por otro lado, las claves de reCaptcha V2 no son válidas para reCaptcha V3.

Hasta el momento de escribir este artículo, el autor de uno de los plugin de formulario más utilizado, no ha actualizado el código para resolver esta incompatibilidad.

También merece comentarse, la experiencia con el reCaptcha V3 vivida en nuestros servidores: El uso de esta nueva versión ralentiza considerablemente el servidor, ya que está continuamente llamando a su script, no sólo cuando se está en un formulario, sino en todo momento.

WordPress o Linux

Sorprendente, ¿verdad? Si, yo también me quedé sorprendido cuando un amigo me hizo esa pregunta:

«Quiero poner una página web y no sé qué hacer, si contratar un servidor Linux o un servidor WordPress, ¿Qué me aconsejas?»

WordPress o Linux
En un principio, lo primero que me vino a la cabeza fue decir que no tiene nada que ver un servidor con un sistema operativo (en este caso Linux) con que tenga instalado o no WordPress, que puede instalarse en Linux o en IIS de Microsoft ®

Luego recordé que para el usuario final, esto debe ser transparente y, por tanto, la pregunta re-formulada debe ser: ¿debo contratar un servicio de alojamiento especial para WordPress o sólo con Linux?
Puede que parezca lo mismo, pero no es igual, hay mucha diferencia entre un alojamiento de WordPress y uno de Linux, y todo depende del uso que quieras dar al servidor. Intentaré explicarlo rápidamente.

Alojamiento WordPress:
Ventajas
- La ventaja más significativa de un alojamiento dedicado de WordPress es la velocidad. Al menos sobre el papel, ofrecen velocidades de 1 o 2 segundos menos.
- El soporte técnico. Cuando se contrata este tipo de servicio, suele haber un departamento técnico especializado que puede resolver tus dudas o problemas con WordPress.

Desventajas:
- Cualquier cosa que requiera una intervención en el servidor, es posible que no se pueda realizar por estar «fuera del rango de acceso»
- Si quieres hacer intervenciones en las bases de datos SQL o en el PHP, normalmente has de contar con la colaboración del servicio técnico.
- Aunque no es muy habitual, el servidor dónde está alojado el WordPress puede ser un IIS de Microsoft ®, por lo que el funcionamiento, si quieres realizar tareas tan sencillas como una copia de seguridad, ármate de paciencia.

Alojamiento Linux:
Ventajas
- Puedes montar el software que te apetezca, sea creado por ti mismo en PHP, Ruby, Java, etcétera o, un CMS o, un CRM o, cualquier otra aplicación.
- Al contratar, puedes solicitar la instalación del software de tu elección y, el «motor» que elijas para base de datos (MySQL, MariaDB, Mongo, etcétera), para servir la web (Apache, NGinx), el lenguaje (PHP, Ruby, Perl …)

Desventajas:
- Aunque exista un soporte técnico por parte del servicio de alojamiento, normalmente se entiende que sabes lo que haces, por lo que deberás contar con mayores conocimientos técnicos.
- Existe un largo número de acciones a realizar para mantener en plena forma el servidor y que tu página web sea rápida, segura, sólida.

Sin embargo, hay muchos planes de alojamiento (dependiendo del proveedor) que ofrecen paquetes Linux y paquetes WordPress.
Lo cierto es que no hay diferencia entre los dos, a nivel del servidor o sus prestaciones, en la mayoría de los casos, cuando ofrecen el paquete de WordPress, no es más que un servidor con un Linux optimizado para su uso con el CRM
Esa optimización consiste en un aumento del valor de imput_vars más que nada por que lo requieren algunas templates, y en algunos casos, se instala un sistema de cache para dar un poco más de agilidad.
Incluso en muchos servicios de alojamiento, si comparamos los planes que ofrecen nos damos cuenta de que «la máquina» es la misma, incluso el precio es el mismo, aunque algunos tienen un precio más alto para el paquete WordPress, por el «servicio técnico especializado».
Así que, desde este punto de vista, la diferencia entre un servidor Linux y un servidor WordPress es: **¡Ninguna!**

¡Ah! Por cierto, si estás interesado en un servicio de alojamiento para tu WordPress, usa el formulario de contacto y dinos lo que buscas, ¡hablamos tu mismo idioma!

Los keywords para encontrar tu sitio

¿Quieres saber qué términos de búsqueda usa la gente para encontrar tu sitio?

Los Keywords son las frases (o palabras) que la gente usa en los buscadores (Google, Bing, DuckDuckGo, …) para encontrar lo que estén buscando por ejemplo, tu sitio.

Quizá por curiosidad o, quizá por que te interesa saber cómo se posiciona tu sitio y así poner las palabras adecuadas para que tu sitio esté uno de los primeros en la lista.

Te voy a contar cómo saber fácilmente, que Keywords ha usado la gente para encontrar tu sitio de WordPress.

¿Qué es el Keyword Tracking y porqué es importante.
Keyword Tracking es la forma en que se conoce el seguimiento de términos clave; básicamente se trata de la actividad de monitorizar la posición de tu sitio con respecto a otros.

Esta actividad te ayuda a ver métricas importantes cómo los Keywords específicos que usa la gente para encontrar tu sitio de WordPress, de esa forma, puedes centrar tu esfuerzo en lo que funciona.

En lo que respecta a la optimización SEO de WordPress, es a menudo recomendable realizar búsquedas para adecuar los meta datos de tu sitio (como la descripción).
Lo que muchos principiantes no saben es que los algoritmos usados por las empresas de búsqueda, cambian a menudo. Si una empresa nueva entra en tu sector, o si tu competencia actualiza su posicionamiento, esa posición que tanto te ha costado ganar, está seriamente en peligro.

Perder posiciones en la lista (perder posiciones en el «ranking») es perder tráfico y quizá clientes.

Los Keywords que la gente usa para encontrar tu sitio
La mejor manera de hacer el seguimiento de los Keywords que usa la gente y los que han permitido que tu sitio esté en la lista (en el «Ranking») es usar la herramienta Google Search Console (Consola del Buscador de Google).

Google Search Console es una herramienta gratuita para que los propietarios de sitios web puedan monitorizar y mantener la presencia de su sitio en los resultados de Google.

Quiero mostrarte cómo conectar la consola del buscador con tu Google Analytics y cómo mostrar esos resultados en el escritorio de WordPress.
Además veremos cómo ver no sólo tus Keywords sino los que han usado en tu competencia para posicionarse.

Si te parece, ¡empezamos!
Hacer el seguimiento de tus Keywords en Google Search Console
Si no lo has hecho aún, debes añadir tu sitio a la herramienta Google Search Console. Un tutorial detallado, llegará pronto.

Para hacerlo de forma fácil y rápida, puedes seguir la guía que está en la herramienta, pero está en inglés.
Una vez que tengas tu sitio identificado con Google Search Console, podrás usarlo para monitorizar tu posicionamiento.

Para ver el posicionamiento de tu sitio, haz clic en el informe de rendimiento («Performance») y luego en clasificación media («average position score»).

La herramienta cargará tus informes, incluyendo la columna de «average position»

A continuación tienes que desplazarte un poco hacia abajo para ver la lista completa de términos por los que está posicionado tu sitio.

Ahí puedes ver una lista de palabras clave con un número de clics, impresiones y, la posición de esa palabra clave en los resultados de búsqueda.

Podrás ordenar los datos por número de clics, impresiones y, la posición (las cabeceras de las columnas).

Al ir descendiendo, podrás ver los Keywords con los que tu sitio se sitúa más abajo en la tabla de resultados. Podrás usar esto como información de los Keywords que tienes que cambiar.
Como siempre, existen plugins para los que no quieren complicarse mucho. El método alternativo es:

Analiza tus Keywords dentro de WordPress con MonsterInsights
Para hacer las cosas de este modo, hacemos uso de un plugin llamado «MonsterInsights» que realiza la búsqueda en Google Search Console y la presenta en el escritorio.

Este método tiene dos ventajas:
1. Puedes ver tus datos directamente en el escritorio de administración de tu WordPress.
2. Se presentarán otros informes generados por «MonsterInsights» que te ayudarán a planear tu posicionamiento más eficientemente.

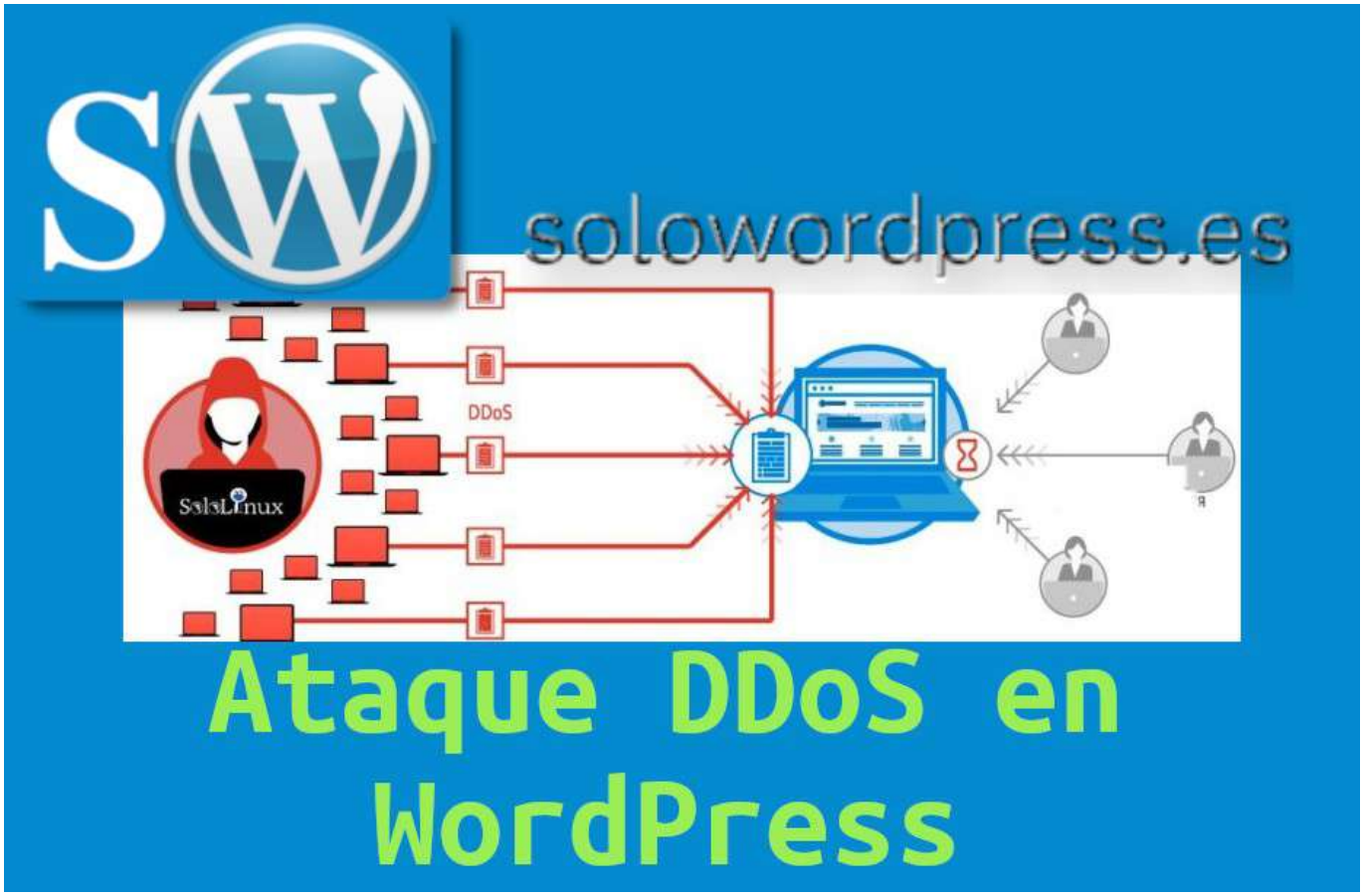
Este plugin es quizá el número uno en WordPress analizando el Google Analytics; te permite instalar fácilmente la herramienta en tu WordPress y te muestra informes que un humano puede entender, en tu escritorio.
Obviamente lo primero que debes hacer, es instalar el plugin y activarlo.
El siguiente paso ha de ser acceder a tu cuenta de Google Analytics, hacer clic en el botón de administración, luego en «All products» en la columna de propiedades y, finalmente en el botón «Link Search Console».
Con esto entrarás en la página de ajustes de la Consola, haz clic en el botón Añadir («Add») y encontrarás una lista de los sitios que tienes enlazados con tu cuenta de Google Search Console.
Entra la url de tu sitio en el campo correspondiente y haz clic en el botón «OK», con lo que tu sitio estará enlazado.
Ahora podrás ver los keywords por los que está listado tu sitio en el ranking, en el área de administración.
Sólo nos queda ir al menú Insights -> Informes y hacer clic en la pestaña Search Console.
Ahí verás una lista de keywords con los que aparece tu sitio en los resultados de búsqueda; junto a ese, verás los siguientes parámetros:
- **Cliks** – Qué tan a menudo es accedido tu sitio buscando ese término.
- **Impressions** – Cuán a menudo aparece en los resultados de búsqueda por ese keyword.
- **CTR** – Click Through Rate para este término.
- **Average position** – La posición media de tu sitio en los resultados de búsqueda por ese keyword.

Método 3 – Hacer el seguimiento de tus Keywords en Google Analytics
Si no quieres tener la información en el escritorio de tu WordPress, puedes ver la información en Google Analytics
Para esto, simplemente accede a tu cuenta de Google Analytics y dirígete a Acquisitions -> Search Console -> Queries report.
Los keywords aparecerán listados en la columna con la cabecera «**Search Query**», junto con el correspondiente «**CTR**», «**impressions**» y «**Average position**».

Hacer el seguimiento de los Keywords de tu competencia usando SEMRush
Si quieres hacer el seguimiento no sólo de los keywords de tu sitio, sino también los de la competencia, veamos cómo hacerlo y, además, superar a tus competidores.
Para este método, usaremos SEMRush; Es una de las herramientas de SEO más usadas, porque te ayuda a atraer más tráfico a tu sitio.
Lo primero que hay que hacer es crear una cuenta en SEMRush.
A continuación, en el escritorio de SEMRush, introduce la url de tu sitio en la barra de búsqueda de la parte superior.
SEMRushte mostrará un completo informe de los keywords más altos en el posicionamiento.
Puedes hacer clic en el botón **View Full Report** para ver la lista completa de keywords.
Junto con cada keyword, verás la posición, volumen de búsqueda, coste (para anuncios pagados) y el porcentaje de tráfico que se dirige a tu sitio.
Puede entrar la url de tu competencia para obtener un informe de los keywords que utiliza y su posicionamiento.

Un par de consejos
Al analizar la lista, encontrarás algunos resultados bien posicionados (menos del puesto 10) con un «impressions» significativo o un «CTR» muy bajo.
Eso quiere decir que tu visitante no encontró tu artículo suficientemente interesante. Puedes cambiar eso cambiando el título del artículo o la meta descripción.
Verás también algunos keyword en los que tu sitio puede fácilmente obtener mejor posición en el ranking. Para mejorar esto, edita el artículo en cuestión y agrega algunas palabras de ayuda o, un vídeo descriptivo e intenta hacer el artículo más fácil de leer.
Si estás usando SEMRush, puedes usar su Asistente de Escritura, que te ayuda a mejorar tu escritura haciéndola más amigable con el SEO para ese keyword en concreto.

Detener y prevenir ataques DDoS en WordPress



WordPress es el CMS quizá más popular y usado del mundo y, no sólo se usa para crear bitácoras, también para todo tipo de páginas web.

Al ser uno de los más populares, es también uno de los más atacados. Como sabes, WordPress está alojado en un servidor y por tanto, es un candidato para sufrir ataques por parte de «los malos», también al servidor.

Los ataques DDoS pueden ralentizar los servidores y, eventualmente, hacerlos inaccesibles por los usuarios o visitantes.

Los ataques pueden, y ciertamente se producen, en servidores de cualquier tamaño.

Si usas un servidor con Linux, te interesa leer los artículos que sobre este tema están publicados en nuestra «web madre» https://sololinux.es, por ejemplo: https://www.sololinux.es/diferentes-tipos-de-ataques-ddos/, en el que se explican los diferentes tipos de ataques DDoS.

Puede que te preguntes ¿cómo puede un negocio pequeño que usa WordPress prevenir un ataque DDoS con pocos recursos?

Eso es exactamente lo que te voy a contar, para que no dependas únicamente de las medidas de seguridad de tu servidor.

¿Qué es un ataque DDos?
DDoS son las siglas del tipo de ataque Distributed Denial of Service, o sea, [ataque] Distribuido de Denegación de Servicio. Es un tipo de ciber ataque que hace uso de ordenadores y dispositivos comprometidos, para enviar o solicitar datos a un servidor (con WordPress, en nuestro caso).

El propósito de los requerimientos es frenar y hasta detener el servidor escogido como objetivo.

Este tipo de ataque se considera una evolución de los ataques DoS Denial of Service, siendo la diferencia el factor de multiplicidad, usando muchos ordenadores (en puntos distintos del planeta) atacando simultáneamente un mismo objetivo.

En ocasiones, estos dispositivos comprometidos se consideran una misma «red» que se denomina **botnet**, ya que cada máquina afectada actúa como un robot y dirige su ataque al sistema objetivo.

El factor distribución permite, en cierta medida, pasar desapercibido por un tiempo y así causar el mayor daño posible antes de ser detectado.

Incluso las grandes empresas y corporaciones internacionales, son susceptibles de sufrir un ataque DDoS.

Este tipo de ataques no son, necesariamente, para robar información de un servidor.

En 2018, la compañía GitHub, una popular plataforma de compartición de documentación, sufrió un ataque DDoS en el que le enviaron 1,3 terabytes por segundo al servidor.

Muchos otros grandes de Internet han sufrido ataques DDoS, por ejemplo: Amazon, Netflix, PayPal, Visa, AirBnB, The New York Times, Reddit, o DYN.

¿Por qué ocurren los ataques DDoS?
Existen varios motivos, las motivaciones más comunes son:
- Gentes con conocimientos técnicos que se aburre y quiere una aventura.
- Personas o grupos que quieren reivindicar su punto (generalmente político).
- Grupos cuyo objetivo es una región o país y sus servicios .
- Grupos o personas cuyo objetivo es una empresa específica, para causarle un perjuicio económico.
- Por extorsión o secuestro (se conoce como «blackmail» o «ransomware»

¿Es lo mismo un ataque por fuerza bruta que un DDoS?
Por lo general, los ataques por fuerza bruta, están dirigidos a entrar en un servidor usando la cuenta de un usuario, adivinando la contraseña o probando con contraseñas al azar, hasta conseguir acceso.

Los ataques DDoS, son ataques cuyo objetivo no es acceder al servidor, sino ralentizarlo o hacer que nadie pueda acceder.

¿Qué daños puede causar un ataque DDoS?
Quizá uno de los ataques más peligrosos para un negocio. El ataque DDoS ralentiza o incluso detiene el servidor objetivo, eso se traduce en que tus clientes o potenciales clientes, no puede acceder a tu web, lo que les disuade de hacer negocio contigo.

No sólo pierdes un cliente, quizá muchos negocios con ese o muchos más clientes y el coste de recuperación puede ser muy alto.

Se me ocurren muchos puntos de difícil cálculo de su costo:
- La pérdida de uno o más negocios.
- El tiempo y esfuerzo invertido en atención al cliente para explicar porqué la interrupción del servicio.
- El coste que puede suponer contratar expertos en seguridad que «arreglen» el problema.
- Pero el mayor de todos, será la mala experiencia sufrida por los usuarios actuales.

Y ahora, vamos al grano.

¿Cómo detener y prevenir un ataque DDoS en WordPress?
Los ataques DDoS pueden ser muy difíciles de detectar, si están «inteligentemente» disfrazados. Sin embargo, hay algunas prácticas de seguridad básicas que pueden ayudar en su prevención y detención.

Elimina los vectores verticales de ataque
Suena extraño, ¿verdad? No es para asustarse.

La gran ventaja de WordPress es que es muy flexible. Esa flexibilidad nos permite integrar herramientas y plugins para otorgarle mayor robustez.

WordPress tiene varias API disponibles para los programadores, con las que los nuevos plugins y servicios se pueden integrar.

Dentro de las medidas que podemos tomar para mitigar los efectos de los ataques hasta incluso prevenirlos están:

Deshabilitar el servicio XML RPC en WordPress
XML-RPC permite que apps de terceros se comuniquen con WordPress. Por ejemplo, necesitarás XML-RPC para usar la app de WordPress en tu teléfono (celular).

Si eres como la gran mayoría, que no usa esa aplicación en el teléfono móvil, puedes deshabilitar el uso de RPC desde el servidor.

Para eso, debes editar el archivo '.htaccess' de tu servidor e incluir el siguiente código:

```
# Block WordPress xmlrpc.php requests
<Files xmlrpc.php>
order deny,allow
deny from all
</Files>
```

Adicionalmente, te sugiero que deshabilites XML-RPC dentro de WordPress. Para eso, tienes que añadir una línea de código en tu plugin personal. Si no recuerdas cómo hacer tu plugin, visita: Mi Primer Plugin en WordPress. El código sencillo es:

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

Deshabilita la REST API de WordPress
La aplicación de WordPress JSON REST API permite a los plugin el acceso a los datos de WordPress, actualizar contenido, e incluso borrarlo. Para deshabilitar este servicio, lo más sencillo es instalar y activar un plugin llamado **Disable WP Rest API**.

Este plugin deshabilita la API para todos los usuarios, desde el momento que lo actives.

Instala un Firewall
Independientemente de si tu servidor tiene instalado un firewall (que debería) puedes instalar lo que se conoce como firewall de aplicación (WAF – Website Application Firewall). En este caso, un plugin de Firewall para WordPress.

La razón para esto, es que un firewall en el servidor seguramente es «genérico», mientras que los plugin están diseñados específicamente para ser usados con WordPress.

Aunque hayas bloqueado los ataques por XML-RPC y REST API, hay otros peligros que un firewall te ayudará a combatir de forma más cómoda.

Cosas tan tontas como activar el bloqueo de una IP en particular o una serie de IPs, son mucho más fáciles si las haces con un plugin, en lugar de a mano, una a una.

Instalar un WAF, te protegerá en gran medida, pero ten en cuenta que cuando el Firewall empiece a actuar, el ataque DDoS ya habrá llegado a tu servidor, por lo que puede ser un poco tarde.

Un servicio externo que quizá es recomendable instalar (aunque tiene un coste) es el de **Sucuri**, que actúa a escala de DNS, por lo que puede detectar y bloquear un ataque DDoS incluso antes de que llegue efectivamente a tu servidor.

¿Cómo diferenciar un DDoS de un ataque de Fuerza Bruta?
Por desgracia, a simple vista, ambos ataques hacen uso intensivo de los recursos del servidor, por lo que los síntomas pueden ser muy parecidos.

Si has instalado Sucuri u otro plugin de seguridad, podrás ver en los informes que ofrece el plugin sobre el acceso, la cantidad de intentos, lo que es un indicativo de un ataque por Fuerza Bruta.

¿Qué hacer si te encuentras en un ataque DDoS?
Los ataques DDoS pueden aparecer incluso si estás protegido con un firewall y las demás protecciones que he mencionado. Las compañías como Sucuri o ClouFlare, están continuamente monitorizando y comparando con sus bases de datos y sus recursos.

Por esa razón, si tienes un plan contratado con alguno de ellos, es probable que ni siquiera te percates de que «los malos» han lanzado un ataque contra tu sitio, ya que ellos mitigan rápidamente sus efectos.

Sin embargo, en algunos casos en que los ataques son de gran envergadura, es posible que el impacto si llegues a sentirlo. En ese caso, mejor si te «pilla preparado».

Algunos consejos para minimizar el impacto de un DDoS:

Alerta a los miembros de tu equipo
Si tienes un equipo de colaboradores (autores, editores, administradores, etcétera) informales rápidamente del incidente. Así tendrán la oportunidad de prepararse para las quejas de tus clientes y ayudar en lo posible.

Avisa a tus clientes sobre el incidente
Un ataque DDoS impactará seriamente la experiencia de tus visitantes. Si tienes una tienda (por ejemplo con WooCommerce) tus clientes no podrán hacer pedidos ni compras y, posiblemente, ni siquiera acceder a sus cuentas.

Utiliza tus redes sociales para comunicar el incidente a tus seguidores y que sepan que volverás a estar en servicio en breve. No es una vergüenza sufrir un ataque informático.

Si la duración del incidente es larga, puedes usar el correo electrónico para comunicar a tus clientes y que estén pendientes de las redes sociales donde avisarás del restablecimiento del servicio.

Si algunos de tus clientes son VIP, usa el teléfono y comunicales personalmente el incidente y que sepan que ya estás trabajando en la solución.

Mantener una comunicación activa, mantendrá fuerte tu reputación al dar seriedad a tu marca.

Contacta con los servicios de seguridad y alojamiento.
Contacta rápidamente con tu servicio de alojamiento. El ataque que estás sufriendo puede ser parte de un ataque de mayores dimensiones, atacando a todos los servicios de tu proveedor. En ese caso, te darán información del estado de la situación.

Si has contratado los servicios de un Firewall externo, contacta con ellos para que sepan que estás sufriendo un ataque; ellos podrán guiarte y ayudarte, incluso quizá mitigar completa o parcialmente el impacto del ataque.

Mantener tu WordPress seguro
WordPress es suficientemente seguro, nada más instalarlo, sin embargo, es recomendable asegurar todo lo posible, recuerda que al ser su uso tan popular, también es popular para «los malos».

Afortunadamente, hay acciones que puedes realizar para mantenerte aún más seguro y, aunque nunca se está preparado para todas las contingencias, muchas si son «controlables».

Recuerda, mantente actualizado, realiza copias de seguridad regularmente y recuerda, el eslabón más débil es usualmente el factor humano.

Ocultar la pagina de inicio de sesión de WordPress

Cambiar la URL de inicio de sesión de WordPress y ocultar el `wp-admin` para burlar a los hackers de sombrero negro y evitar ataques de fuerza bruta … ¡es más fácil de lo que crees, hacer que tu sitio sea más difícil de descifrar!

No nos engañemos. Incluso los novatos saben que todo lo que tienen que hacer para hacer que la vida del propietario de un sitio de WordPress sea miserable es encontrar la página de inicio de sesión de WordPress y adivinar el nombre de usuario y la contraseña.

Adivinar contraseñas, por cierto, no es difícil, especialmente si usas las mismas contraseñas para la mayoría de tus inicios de sesión y compartes toda tu vida en las redes sociales.

WordPress es probablemente, el `CMS` más popular del mundo y esto lo convierte en un imán irresistible para los piratas informáticos y los intentos de inicio de sesión maliciosos.

Incluso lo mejor de lo mejor puede ser derribado por un disidente sigiloso con acceso a herramientas de fuerza bruta que automáticamente intentarán adivinar tu nombre de usuario y contraseña presionando tu página de inicio de sesión de WordPress una y otra vez.

La mejor manera de luchar contra los ataques de fuerza bruta (que así se conoce esta técnica) es … ¡Esconderse!

Los intentos de fuerza bruta para iniciar sesión en WordPress son tan comunes que incluso hay una página en el Codex dedicada al tema.

Pero … ¿por qué los piratas informáticos y los robots maliciosos tienen la oportunidad de intentar adivinar tus datos de inicio de sesión? Simplemente oculta tu página de inicio de sesión de WordPress y la mayoría de los bots y el software automatizado ni siquiera sabrán que tu sitio existe.

Lo que quiero es que aprendas cómo implementar una de las estrategias más simples y fáciles para proteger tu sitio de **crackers** y **bots maliciosos**: cambiar la URL de inicio de sesión de WordPress, ocultar tu página de inicio de sesión de `wp-admin` y `wp-login` y redirija a los visitantes no deseados lejos de tu página de inicio de sesión.

Déjalo abierto y los crackers piratearán. Oculta la página de inicio de sesión de WordPress y … ¡sin ataques maliciosos!

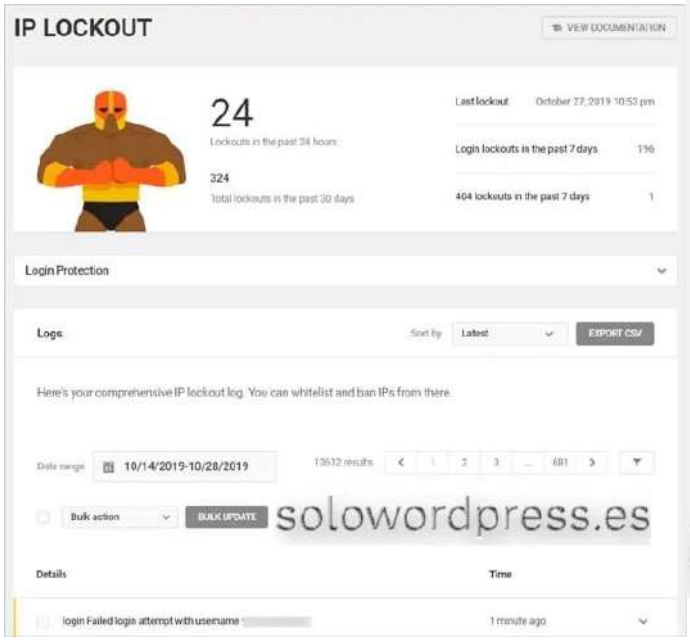
¿Por qué cambiar la URL de inicio de sesión de WordPress?
Tengo un sitio estándar de WordPress que instalé hace unos años. Para acceder a la página de inicio de sesión, todo lo que tiene que hacer es ir a `wp-admin` o `wp-login`.

Este sitio no ve mucho tráfico. En un mes típico, genera alrededor de 5,000 páginas vistas. Sin embargo, la página de inicio de sesión del sitio ve intentos de inicio de sesión maliciosos de manera sorprendentemente regular.

Tengo un plugin de seguridad activado y rastrea el número de intentos de inicio de sesión maliciosos bloqueados. Desde su instalación, puedo ver que mi sitio maneja cientos de intentos de inicio de sesión maliciosos cada mes, con un promedio de aproximadamente 20 al día, o un intento de inicio de sesión malicioso cada 60 minutos.

Los intentos de inicio de sesión no ocurren uno por hora. Pueden pasar semanas sin que se registre un solo intento de inicio de sesión malicioso. Luego, de repente, se registrarán unos cientos o incluso un par de miles de intentos de inicio de sesión en un corto período.

La mayoría de los sitios de WordPress configurados como instalaciones estándar experimentan periódicamente ataques de fuerza bruta al intentar iniciar sesión en el panel de WordPress. El tuyo probablemente también, lo sepas o no.



Los piratas informáticos pueden determinar fácilmente si un sitio funciona con WordPress o no (a menudo simplemente mirando la fuente de la página).

Una vez que un **crackers** sabe que tu sitio se ejecuta en WordPress, también sabe cómo encontrar tu URL de inicio de sesión de WordPress (alerta de spoiler: la URL de inicio de sesión predeterminada de WordPress se encuentra ingresando tu nombre de dominio, seguido de `/wp-login.php`).

El comportamiento predeterminado de WordPress carga la página de inicio de sesión cuando accede a `wp-login.php`. Si en su lugar escribes `wp-admin`, serás redirigido automáticamente a `wp-login.php`.

A menos que sepas cómo cambiar tu nombre de usuario administrador, tu amigable pirata informático vecino también sabrá que tu nombre de usuario probablemente sea algo así como **administrador** o **admin**.

Todo lo que el **cracker** tiene que hacer ahora es adivinar la contraseña. Incluso si no pueden adivinar la contraseña pero siguen intentándolo, esto puede agotar los recursos de su servidor y posiblemente «tirar abajo» tu sitio.

Si los piratas informáticos acceden ilegalmente a tu inicio de sesión el tiempo suficiente, probablemente generarán suficientes visitas para adivinar su contraseña.

Si no pueden verlo, no pueden descifrarlo
Muchos piratas informáticos son oportunistas y buscan las cosas fáciles; si dejas tu billetera sobre la mesa en una cafetería, los oportunistas la cogerán.

Si no quieres que los oportunistas te roben, mantén ocultas tus pertenencias.

Continuando con esta analogía, tu página de inicio de sesión de WordPress brinda a los usuarios administradores acceso a todo tu dinero, así que como parte de nuestra estrategia de crear «seguridad a través de la oscuridad», ocultamos la URL de tu página de inicio de sesión a todos excepto al administrador.

Opcionalmente, instala WordPress en su propio directorio
Ya sea por que se trate de una nueva instalación de WordPress o de un sitio web existente, siempre que sea posible, considera instalar WordPress en un subdirectorio. Si bien esto no evitará que los piratas informáticos encuentren tu página de inicio de sesión de WordPress si eligen deliberadamente apuntar a tu sitio, desalentará a muchos «bots» aleatorios y usuarios maliciosos que buscan objetivos fáciles para comenzar a golpear tu sitio y sacudir tu árbol para ver lo que cae.

Tener tu sitio de WordPress instalado en un subdirectorio, es un buen primer paso para crear «seguridad a través de la oscuridad».

Como siempre, antes de hacer cualquier otra cosa, crea una copia de seguridad completa de tu sitio y almacenala en un lugar donde no la borres o modifiques accidentalmente.

Una cosa más. Al crear un subdirectorio, elije un nombre que no sea demasiado predecible como `https://ejemplo.com/wordpress` o `http://ejemplo.com/wp`. En su lugar, elije algo único que nadie podrá adivinar como `https://ejemplo.com/ddiwp` (un acrónimo de directorio donde instalé WordPress).

El siguiente paso es ocultar la URL de tu página de inicio de sesión (y opcionalmente redirigir a los visitantes de `wp-login.php` a otra página de tu sitio).

Hay algunas formas en que puedes ocultar tu página de inicio de sesión de WP a otros usuarios:
- Usa un plugin para enmascarar tu URL de inicio de sesión (la forma más fácil)
- Enmascara tu URL de inicio de sesión de WordPress sin un complemento (para los geek)
- Modifica el archivo .htaccess.

Ocultar la página de inicio de sesión de tu sitio – Descargo de responsabilidad
Antes de comenzar, ten en cuenta que esta estrategia no es recomendable si se requiere una página de inicio de sesión que debe ser fácil de encontrar para otros usuarios (como un sitio de membresía).

Si tu sitio no es un sitio de membresía y los intentos de inicio de sesión se limitan a una docena o menos de administradores, autores, editores y colaboradores, entonces ocultar tu página de inicio de sesión ayudará a proteger tu sitio contra intentos de inicio de sesión maliciosos.

Ocultar wp-login.php usando un plugin
Hay una serie de plugins gratuitos de WordPress que te permitirán ocultar la URL de la página de inicio de sesión. Algunos de estos complementos también te permitirán redirigir a los visitantes de wp-login.php a otra página de tu sitio web.

Dirígete al menú `Plugins -> Añadir nuevo` y busca «Ocultar inicio de sesión» para ver una lista de complementos de seguridad que permiten hacer eso.

Para este tutorial, utilizaremos el complemento Defender de WPMU DEV, pero insisto, hay muchos.

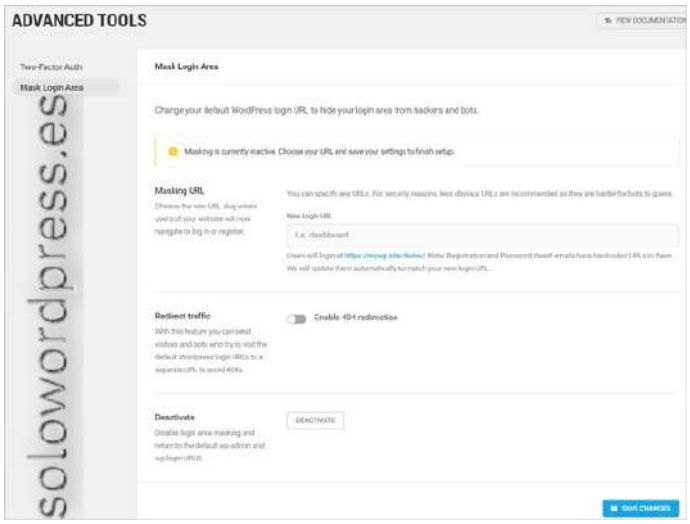
Defender te permite ocultar y redirigir `wp-login.php`, e incluye muchas otras características de seguridad.

Después de instalar y activar el complemento, navega hasta el menú principal del panel de WordPress y ve a `Defender -> Dashboard`.

Localiza la sección «`Mask Login Area`» y haz clic en el botón «`Activate`» para activar la función.

Haz clic en el botón «`Finish Setup`» para abrir la pantalla de opciones de enmascaramiento de URL.

Esto abre la pantalla Herramientas avanzadas (`Advanced Tools`).



Debes ingresar, en la sección «`Masking URL`», una nueva dirección donde los usuarios de tu sitio irán para iniciar sesión o registrarse. Una vez más, recomiendo elegir algo que puedas recordar fácilmente, pero todos los demás no puedan adivinar al azar.

Para este ejemplo, usaré el mismo método de acrónimo utilizado anteriormente para encontrar el nombre del directorio `ddiwp` y nombré nuestra nueva URL de inicio de sesión de WordPress algo único como:

`http://ejemplo.com/ddiwp/gli`

En este caso, `gli` significa iniciar sesión y cumple el objetivo de ser simultáneamente fácil de recordar y difícil de adivinar.

De esta forma, conseguimos que tu nueva URL de inicio de sesión de WordPress sea difícil de adivinar para los piratas informáticos.

Guarda los cambios y cierra sesión en tu sitio de WordPress.

Ahora, intenta iniciar sesión nuevamente a través de la página de inicio de sesión predeterminada en `yourdomain.com/wp-login.php`.

Normalmente, escribir `wp-admin` en un navegador web redirige automáticamente a los usuarios a `wp-login.php`. Defender también deshabilita esta característica.

Solo los usuarios con acceso a la URL enmascarada verán la página de inicio de sesión de WordPress.

La URL de tu página de inicio de sesión de WordPress ahora está enmascarada, para iniciar sesión deberás entrar la URL: `http://ejemplo.com/ddiwp/gli`

Como un toque extra agradable para tus usuarios, puedes personalizar tu página de inicio de sesión de WordPress (con Mi primer plugin de WordPress), instalar plugins para mejorar el inicio de sesión y el registro del usuario, o permitir que los usuarios inicien sesión en WordPress utilizando una dirección de correo electrónico. Sin embargo, si solo ciertos usuarios pueden acceder a tu sección de administración, puedes limitar el acceso a la página de inicio de sesión para usuarios específicos por direcciones IP.

Paso opcional: Redirigir wp-login.php

Usando el método anterior, cualquier persona que intente visitar la página de inicio de sesión predeterminada de WordPress (es decir, `wp-login.php`) recibirá un mensaje de error («Esta función está desactivada»).

Si deseas enviar visitantes y usuarios (o incluso piratas informáticos) a una página diferente (por ejemplo, la página de tu tienda, página de contacto, sección de preguntas frecuentes o cualquier otra página de tu sitio), puedes redirigir la URL predeterminada de `wp-login.php`, usando la función de «`Redirect traffic`» de Defender.

Para redirigir la página `wp-login.php`, debes ir al menú `Defender -> Advanced Tools -> Mask Login Area`.

Habilita la redirección 404 en la sección «`Redirect traffic`», ingresa el «**slug**» de la página a la que deseas enviar visitantes y haz clic en el botón «`Save Changes`» para actualizar la configuración.

Notas:

1. Tu «slug» puede consistir en cualquier combinación de a-z y 0-9.
2. No puedes agregar URL completas (esto evita enviar sus errores 404 a otro dominio).

Ocultar la página de inicio de sesión de WordPress sin un plugin
Si deseas ocultar tu página de inicio de sesión sin usar un plugin (la manera geek), todo lo que necesitas es un editor de texto, acceder a tus archivos de instalación de WordPress mediante la herramienta que tengas designada (FTP, cPanel File Manager, etc.), y luego:

1. Haz una copia de seguridad de tu archivo `wp-login.php` (Yo recomiendo, de paso, hacer una copia de seguridad de todo).
2. Edita el archivo `wp-login.php`, selecciona todo el contenido y copialo al portapapeles.
3. Crea un nuevo archivo de inicio de sesión PHP. El archivo puede tener el nombre que quieras, p.e. entrada-segura.php o, entrada-guay.php, etcétera.
4. Pega el contenido del portapapeles en ese nuevo documento, guarda los cambios y cierra. Alternativamente, abre el `wp-login.php` y guardalo con el nombre que quieras.
5. En tu nuevo archivo, busca y reemplaza cada instancia de la cadena «`wp-login.php`» por el nombre de tu nuevo archivo y guarda el archivo.
6. Si has editado en local el nuevo archivo, subelo al servidor; recuerda subirlo al directorio que has creado para tu copia de WordPress.
7. Elimina el archivo `wp-login.php`.
8. Prueba tu nuevo inicio de sesión, recuerda acceder con la URL que has diseñado.

Si por alguna razón quieres volver al principio, sólo has de restaurar el archivo `wp-login.php` y eliminar el creado por ti.
Cualquiera que visite la página predeterminada `wp-login.php` experimentará un error.

Trucos para el .htaccess
Hay maneras de «ocultar» los detalles de inicio de sesión de WordPress utilizando el archivo `.htaccess`. Sin embargo, ocultar su URL de inicio de sesión de WordPress no significa necesariamente ocultar lo demás.

Por ejemplo, echemos un vistazo a lo que sucede cuando añades el redireccionamiento de URL a tu `.htaccess`. Recuerda hacer una copia de seguridad completa de tu sitio antes de realizar cambios en el archivo `.htaccess`.

Oscurecer la pagina de inicio de sesión de WordPress con redireccionamiento de URL
Puedes cambiar la ubicación de tu página de inicio de sesión cambiando el nombre del archivo de inicio de sesión de WordPress, utilizando el módulo `mod_rewrite` en un servidor Apache.

Para hacer esto, añade la siguiente línea al archivo `.htaccess` (*nota: reemplaza «newloginpage» con cualquier alias y cambia la URL de ejemplo.com por tu dominio*):

```
RewriteRule ^ newloginpage $ http://www.ejemplo.com/wp-login.php [NC, L]
```

En este ejemplo, usamos un alias llamado «danzadmalditosdanzad» y volveremos a cargar el archivo `.htaccess` en nuestro servidor:

```
RewriteRule ^ danzadmalditosdanzad $ https://ejemplo.com/ddiwp/entrada-guay.php [NC, L]

# BEGIN WordPress
# Las directivas (líneas) entre `BEGIN WordPress` y `END WordPress` se generan
dinámicamente
# , y solo se deberían modificar mediante filtros de WordPress.
# Cualquier cambio en las directivas que hay entre esos marcadores se sobreescribirán.

# END WordPress
Options All -Indexes
```

Ahora, regresa al sitio e ingresa la nueva URL.

Como puedes ver, el método anterior no oculta la URL de inicio de sesión predeterminada de WordPress, simplemente crea un alias que permite a los usuarios iniciar sesión en su panel de WordPress utilizando una dirección web que es más fácil de recordar que `https://tuejemplo.com/wp-login.php`.

Conclusiones
Idealmente, recomendamos seguir usando un plugin si deseas cambiar la URL de inicio de sesión de WordPress, ocultar las páginas `wp-admin o wp-login.php`, o redirigir a los usuarios fuera de la página de inicio de sesión predeterminada. Jugar con el código puede causar problemas de compatibilidad, ralentizar tu sitio y crear otros problemas.

Sin embargo, se puede ocultar la forma de acceder, modificando reglas del archivo de seguridad `.htaccess` de Apache, si te sientes seguro.

WordPress es un imán para piratas informáticos y robots maliciosos, por lo que es importante comprender las mejores prácticas de seguridad de WordPress e implementar múltiples estrategias de seguridad de WordPress para proteger tu sitio de los piratas informáticos y los ataques de fuerza bruta. Esto incluye seguridad a través de la oscuridad.

Cuando se usa como parte de una estrategia de seguridad más integral, la oscuridad puede ser útil. Sin embargo, como acabamos de ver, simplemente ocultar la página de inicio de sesión de WordPress no es suficiente para garantizar que no verás ningún intento de inicio de sesión malicioso.

A menos que cambies realmente la URL de inicio de sesión de WordPress de tu sitio y redirijas a los visitantes no deseados fuera de páginas como `wp-admin o wp-login.php`, los piratas informáticos y los bots aún podrán encontrar tu página de inicio de sesión e intentar adivinar tus credenciales de inicio de sesión.

Cómo mostrar y diseñar metadatos de publicación en WordPress 5.3

Seguro que alguna vez te has fijado en un artículo y has visto que aparece la fecha en la que se publicó o la categoría a la que pertenece ¿cierto? Esto es lo que se denomina «metadatos de publicación», una parte importante de una bitácora.

Usados correctamente, los metadatos mejoran la experiencia del usuario; supongamos que un visitante de tu bitácora está interesado en un tema en concreto, si por ejemplo, tu bitácora habla sobre fotografía, quizá tienes una categoría sobre objetivos y puede que tu visitante esté interesado en objetivos y no en revelado químico, así, la categoría le ayuda a ver rápidamente lo que necesita y no perder el tiempo (o abandonar tu sitio porque no encuentra lo que quiere).

¿Qué son los metadatos de publicación y cómo pueden ayudar a tu blog?
Los metadatos de una publicación contienen información relevante sobre la misma, como la fecha de publicación, el nombre del autor, las categorías, las etiquetas y las taxonomías personalizadas, etc.

Un tu bitácora es importante que te asegures que los metadatos sean correctos, ya que esta información puede ayudar al visitante a comprender más sobre la publicación y también puede ayudar a aumentar las visitas a tu página al facilitar la navegación de tu sitio.

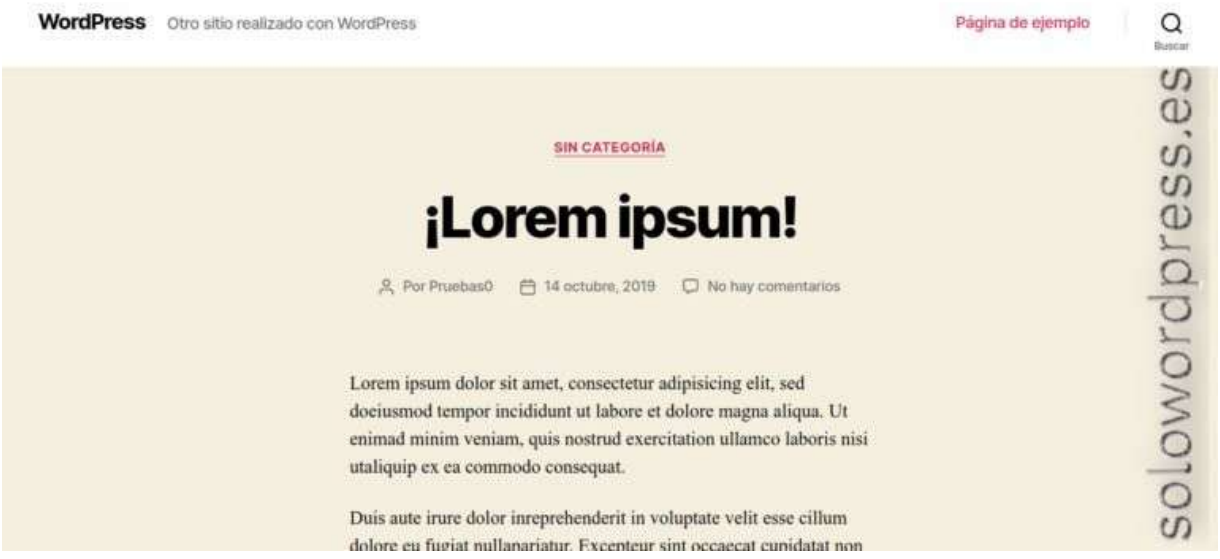
¿Cuántos metadatos de publicación debes mostrar?
La ubicación de los metadatos mostrados en la bitácora varía de un tema a otro. Algunos pueden mostrarlo antes del título de la publicación, algunos después del título y otros justo después del contenido.

Pero demasiadas metadatos pueden estropear el diseño. La situación ideal es aquella en la que solo mostrarías la información que consideres necesaria.

Ahora veamos cómo puedes personalizar y agregar metadatos.

Personalización de metadatos de las entradas (Post)
Como menciono más arriba, la ubicación de los metadatos varía de un tema a otro. Aquí trabajaremos en un tema en particular, en este caso el tema predeterminado en la versión 5.3, el TwentyTwenty, así que ten en cuenta que el código y las páginas pueden ser diferentes en tu tema.

En los temas modernos, los metadatos de las entradas se definen en la página de etiquetas de plantilla y se llaman cuando es necesario, pero en algunos temas es posible que los meta de publicaciones se coloquen directamente antes o después del título de la publicación.



En general, encontrarás metaetiquetas de publicación en `index.php`, `single.php`, `archive.php` y páginas de plantillas de contenido.

Un código simple se vería así:

```
<?php the_permalink(); ?>"><?php the_time( get_option( 'date_format' ) ); ?>
```

Este código mostrará algo como esto: «Cómo hacer buenas fotos> 2030/01/01»

Hoy por hoy, la mayoría de los temas modernos están usando la página de etiquetas de plantilla para manejar los meta (`template-tags.php`). Vamos a ver cómo funciona.

El primer paso es siempre, **crear un tema hijo**, para no cambiar el código del tema padre.

Trabajaremos en una sola página de publicación y así es como se ve desde el front-end.

Para el ejercicio, digamos que queremos agregar un icono del autor antes del nombre del autor, para ello, hacemos lo siguiente:

En el tema TwentyTwenty, analizamos el código del archivo `functions.php`, veremos que está todo claramente organizado, lo que nos ayuda a encontrar dónde se insertan los metadatos (meta-tags).

```
/**
 * REQUIRED FILES
 * Include required files.
 */
require get_template_directory() . '/inc/template-tags.php';
```

Para mantener la integridad, tenemos que mantener la estructura de archivos, por lo que tenemos que crear un directorio «/inc» y poner ahí nuestro archivo `template-tags.php` modificado.

Al editar este archivo, vemos que está documentado con comentarios, la sección que presenta la información del autor de la entrada:

```
// Author.
                if ( in_array( 'author', $post_meta, true ) ) {

                    $has_meta = true;
                    ?>
                    <li class="post-author meta-wrapper">
                        <span class="meta-icon">
                            <span class="screen-reader-text"><?php _e( 'Post author', 'twentytwenty' );
?></span>
                            <?php twentytwenty_the_theme_svg( 'user' ); ?>
                        </span>
                        <span class="meta-text">
                            <?php
                            printf(
                                /* translators: %s: Author name */
                                __( 'By %s', 'twentytwenty' ),
                                '<a href="' .
esc_url( get_author_posts_url( get_the_author_meta( 'ID' ) ) ) . '">' .
esc_html( get_the_author_meta( 'display_name' ) ) . '</a>'
                            );
                            ?>
                        </span>
                    </li>
                    <?php

                }

?>
```

La línea de meta etiquetas está organizada en forma de lista no numerada y vemos que en el espacio correspondiente al autor tenemos dos elementos en forma de etiquetas .

El primero hace una llamada a la función `twentytwenty_the_theme_svg()`, que presentará un icono con forma «humana».

El segundo presenta el nombre del autor del post.

Dado que el ejercicio a realizar es incrustar el icono del autor, lo que hemos de hacer es sustituir el contenido del primer elemento `<span></span>`, por lo que queremos.

Nos dirigimos a nuestro archivo `functions.php` del tema hijo, y agregamos una función que nos permita presentar la imagen del servicio de **«Gravatar»** (es una elección, se puede usar cualquier otro).

O para complicarnos menos, lo hacemos en nuestro propio archivo `template-tags.php` modificado.

Editamos el archivo y sustituimos la línea 348, que es la que presenta la imagen en pantalla, por nuestras líneas

```
<?php $email = get_the_author_meta( 'user_email', 'ID' );
    $default = get_the_author_meta( 'user_url', 'ID' );
    $size = 40;
    $grav_url = "https://www.gravatar.com/avatar/" . md5( strtolower( trim( $email ) ) ) .
"?d=" . urlencode( $default ) . "&s=" . $size;
    ?>
    <img src="<?php echo $grav_url; ?>" alt="Imagen Gravatar de: <?php
esc_html( get_the_author_meta( 'display_name', 'ID' ) ) ?>" />
```

La razón para hacer aquí la construcción de la llamada al servicio de **Gravatar** es que, es aquí donde tenemos toda la información pertinente.

Diseñando los metadatos
Por supuesto, también podemos querer cambiar la forma en que están presentados los metadatos, lo cual es más fácil que cambiar el código PHP del tema.

Para hacer cambios estéticos, sólo necesitamos hacer cambios en el `CSS` asociado y, para eso lo que necesitamos conocer es cómo etiqueta el tema elegido, los metadatos que queremos cambiar.

Si por ejemplo queremos cambiar la apariencia de las categorías que se presentan (categorías a las que pertenece la entrada), sólo tenemos que averiguar el estilo con el que se presentan, lo que se hace fácilmente consultando el navegador.

En este caso,si examinamos el elemento (en Firefox se hace situando el cursor sobre el elemento y haciendo clic derecho, luego seleccionar «**Inspeccionar elemento**» – en la mayoría de los navegadores, esto se consigue también pulsando el atajo de teclado: «**Ctrl+Mayús+C**»).

Aquí vemos que la etiqueta que en este caso aparece, «Sin categoría», tiene asignada una clase llamada `entry-categories` y es precisamente ese estilo el que necesitamos modificar.

Ten en cuenta que al alterar el estilo, lo hacemos de forma genérica, es decir, para TODOS los elementos que tengan esa misma clase.

Para cambiar un estilo definido en el tema padre, lo que tienes que hacer es definirlo de nuevo en el tema hijo, con unas consideraciones:

1. Mientras más específicamente designemos un elemento, más prioridad asigna el navegador a la regla de estilo.
2. Si no asignamos todos los estilos necesarios en nuestra nueva regla CSS, puede que una regla posterior del teme padre, nos deshaga lo estipulado.
3. En algunos casos, es necesario aplicar la directiva `!important` para asegurarnos el cumplimiento de la directiva.

Así que procedemos a incluir la directiva para el formato que queremos alterar en nuestro archivo `style.css` del tema hijo.

Deberemos incluir todas las características que deseemos, así el style.css deberá incluir:

```
.entry-categories-inner {
    font-size: 2rem;
    margin: 1rem 0 0 2.rem;
    border: 1.5rem solid currentColor;
    background-color: #ddd !important;
}
```

Con esto conseguimos que las categorías aparezcan enmarcadas, cada una, en recuadro con borde del mismo color de la letra y fondo gris claro; el tamaño de la letra será dos veces el tamaño normal.

Sorpresa en los anuncios para WordPress 5.4

A medida que nos acercamos a las fechas prometidas para una nueva versión, en este caso la versión 5.4 prometida para el mes de marzo de 2020, empiezan a aparecer noticias, «filtraciones», rumores, etcétera.

Sorpresa en los anuncios para WordPress 5.4
Una de esas filtraciones, que son un secreto a voces, es la inclusión como parte del «core» de WordPress, de lo que se conoce como «**carga diferida**» o «**Lazy Load**».

Durante mucho tiempo se ha hablado sobre este tema y, no se ha obtenido una respuesta clara por parte de las grandes de Internet.

Existen varios plugin para WordPress, que realizan esa función, pero no existe hasta hoy, un comportamiento estándar a ese respecto, por lo que cada plataforma, cada navegador, implementa su propia solución.

Felix Arntz, ingeniero principal de WordPress e ingeniero de programas de desarrollo de Google, anunció un plan para impulsar una función de carga diferida en la plataforma.

¿Qué es la carga diferida?
El concepto de carga diferida permite que una página web se procese sin cargar ciertos recursos hasta que se necesiten. Esto conduce a cargas de página más rápidas y guarda datos en el extremo del visitante.

La carga diferida es particularmente útil cuando se procesan imágenes en la web, y más aún cuando sabemos que no todos los dispositivos cuentan con conexión con un gran ancho de banda.

Adopting the new loading attribute is a great chance for WordPress to lead the way for a faster web overall.

Felix Arntz

«La adopción del nuevo atributo de carga es una gran oportunidad para que WordPress lidere el camino para una web más rápida en general.»

Hay, como digo, diferentes soluciones aportadas a este tema. Por lo pronto, la proximación diseñada por Google y adoptada por los navegadores `Chrome, Edge y Opera`, parece que es la que se va imponiendo, aunque en el juego de la dominación de estándares, no podemos olvidar lo que hagan `Firefox, Brave y Safari`.

Y la batalla se libra en el campo de los navegadores, porque hasta ahora la solución aportada se maneja en Javascript, ya que hay que tener en cuenta el `viewport` (la pantalla, para entendernos) y su tamaño, para saber si la imagen debe o no aparecer.

Todo empieza por añadir el atributo `loading` a un elemento `<img>` o `<iframe>`.

Por ahora todo se basa en que hay que instalar un plugin (disponible en el repositorio oficial) que fuerza el añadido del campo `loading` a todos los elementos que sea necesario

Arntz escribió, en la publicación del anuncio: «Con WordPress habilitando la carga diferida nativa de forma predeterminada, impactaría significativamente el rendimiento y la experiencia del usuario para millones de sitios, sin requerir ningún conocimiento técnico o incluso la conciencia de la carga diferida como concepto».

¿Cómo afecta esto a otros plugin?
Debido a que no todos los navegadores web admiten el atributo `loading`, es posible que los usuarios no quieran descartar automáticamente sus complementos actuales cuando la función aparezca en WordPress. Los usuarios pueden optar por admitir navegadores sin carga lenta nativa por un tiempo.

El código propuesto dentro del complemento `Lazy Loading` intenta detectar si el atributo `loading` existe en una imagen antes de aplicarlo. Esto significa que el código debería funcionar bien con los complementos existentes y evitar conflictos en la mayoría de los casos.

Comentario
Personalmente me pregunto si no sería mejor impulsar el uso de imágenes en formato `webp`. El ahorro en peso de este formato de imagen, redundaría no sólo en una carga más rápida, sino en ahorro en recursos necesarios.

El uso de la tecnología `Lazy Loading` implica el necesario uso de recursos adicionales en los servidores y en los clientes, para obtener un resultado casi nulo, ya que lo único que se consigue es diferir la carga, no existe ahorro de ningún tipo.

SoloWordpress

Esta revista es de **distribución gratuita**, si lo consideras oportuno puedes ponerle precio. Tu también puedes ayudar, contamos con la posibilidad de hacer donaciones para la REVISTA, de manera muy simple a través de **PAYPAL**

AYUDANOS A SEGUIR CRECIENDO

PayPal™ Apoyo con una donación

Síguenos en las Redes: