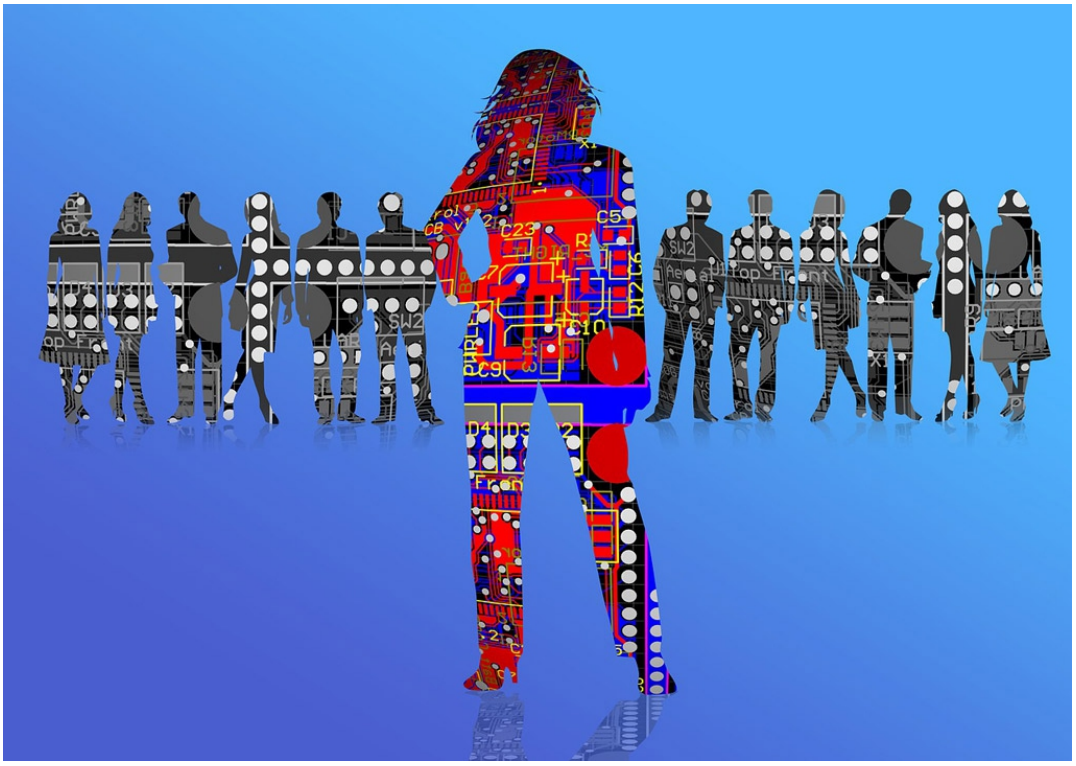
El archivo Robots txt infalible para Wordpress.



El archivo olvidado de Wordpress es sin duda el robots txt, y sin duda es de vital importancia para la correcta indexación de tu web. Además te protege de muchas visitas desagradables y esconde a los ojos de determinados bots los contenidos de tu web. Me atrevería a calificar el archivo robots txt como una de las grandes herramientas imprescindibles para cualquier diseño web y de vital importancia para una buena indexación cuando prepares tu página web para el SEO. Voy a tratar de explicar en este post como programar un archivo robots txt infalible para Wordpress, un archivo que además de protegerte ante miradas poco discretas, ayude a indicar a las arañas de Google y del resto de buscadores donde puede encontrar el contenido relevante de tu web.

Antes de entrar en ver distintos ejemplos de como programar el archivo robots txt para Wordpress, voy a tratar de explicar como funciona.

¿Cómo funciona el robots txt de Wordpress?

WordPress es sin duda un poderoso CMS, sin embargo hay que configurarlo muy bien para evitar crear contenido duplicado.

Vamos a ver un ejemplo de configuración típica de uun diseño web con Wordpress.

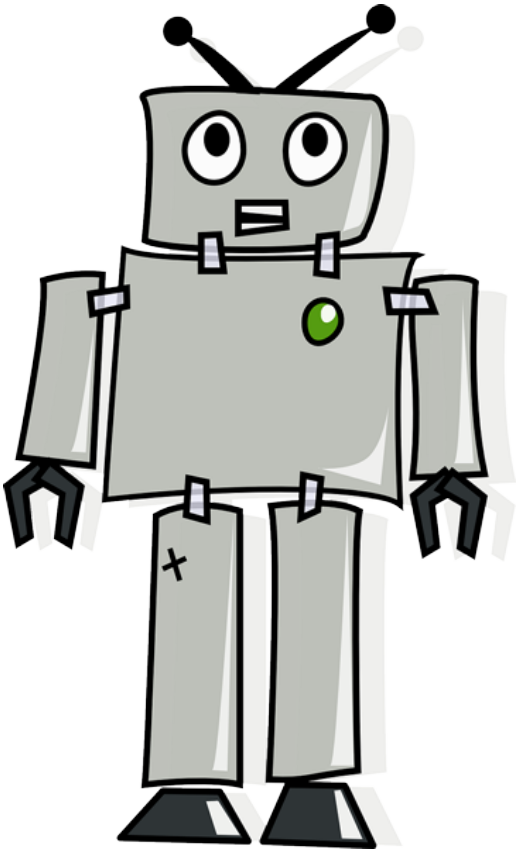
- Home:
  - Contenido destacado y que pone en antecedentes de que vamos a encontrar en la web, normalmente incluye las últimas entradas del blog para que sea fácil acceder a ellas.
- Blog
  - Tu página de entradas, donrán bien visibles todos los post.
- Sidebar:
  - Incluye normalmente una lista con las últimas entradas y una lista de categorías
- Footer:
  - Al igual que en el sidebar, uno de los recursos habituales son las últimas entradas al blog, en muchas ocasiones lo encuentras tanto en Sidebar como en Footer

- Páginas de Categorías
    - Incluyen todas las entradas relacionadas con determinada categoría.
- Página de Author:
    - Incluye las entradas de los autores del blog, casi siempre un sólo autor, y que muestra nuevamente todas las entradas

Ahora procedemos a escribir un post, un magnífico post, con su categoría, sus etiquetas, metadescripciones, metatítulo y las correspondientes keywords, sin embargo no se indexa ni a la de tres, la pregunta es siempre la misma, ¿porqué?

Tu precioso post aparece en la Home, en el blog, en el sidebar, en el footer y en la página de autor, por lo tanto estás mostrando contenido duplicado por todas partes, Si al amigo Panda, la actualización de Google, hay algo que le encanta es precisamente eso, el contenido duplicado para penalizar tu maravilloso diseño web y enviar todo el trabajo de posicionamiento SEO al fondo más oscuro de los resultados de búsqueda. Con una buena programación de nuestro archivo robots txt podemos evitar que este contenido duplicado esté visible para las arañas de Google y así conseguir que nuestros contenidos se indexen mejor y por supuesto, evitar penalizaciones.

Con esta breve introducción vamos a crear nuestro archivo robots txt.



Crear un archivo robots txt para tu web es un proceso sencillo, basta con crear con Notepad, bloc de notas o cualquier otro editor de texto un archivo y nombrarlo robots.txt, subirlo a la raíz de tu FTP y ya que estás puesto se lo envías a Google a través de la **Search Console** para que lo pruebe y lo detecte.

Nota.- Las arañas de Google no siempre respetan el contenido de robots txt, pero eso ya no es culpa del archivo, son las propias arañas que también las gusta investigar porque escondes determinado contenido.

A la hora de penasr como va a ser tu archivo robots txt, también te recomiendo que te des un paseo por los **foros especializados** para ver los nuevos bots malignos que van apareciendo y lo letales que pueden ser para tu web.

Si esto de la programación no va contigo, tambien encontrarás en el repositorio de Wordpress varios plugin que hagan el trabajo por ti, personalmente prefiero personalizar mis diseños web con mis propias herramientas en lugar de usar plugins que ralentizan la carga y el funcionamiento de la web, pero para gustos los colores....

Hay cientos de bots conocidos, los hay totalmente inofensivos y que te ayudan en el posicionamiento, mientras que los hay malos malísimos y que lo único que buscan es usar tu web con no se sabe muy bien que fines, bueno si que se sabe pero eso quizá lo aborde en otro post.

Vamos a ver a nuestros bots favoritos, los buenos:

- Googlebot: necesita presentación? La araña de Google que nos visita de vez en cuando para indexar nuestros contenidos, a mayor y mejor contenido más frecuencia de visita.
- Googlebot-image: esta araña es como su prima para contenidos pero indexa las imágenes
- Roger: me cae bien el bot de MOZ, creo que es de mucha utilidad en SEO
- Yandexbot: del metabuscador ruso Yandex, un imprescindible si quieres posicionar tu web en aquella zona, además ultimamente ha unificado criterios con Google, por tanto no hay que cambiar mucho la forma en que trabajar el SEO para la zona auqneu tiene algunos matices importantes.
- ia_archiver: el bot de Alexa, aunque ha perdido fuelle es muy útil ya que también utiliza el Wayback Machine que te permite ver como lucía una determinada web en el pasado.

Hay muchos más y sin duda te recomiendo que te pases por la **página oficial de robotstxt** para estar al día.

Creando el archivo robots txt

Hay dos reglas fundamentales en todo archivo robots txt, la primera el **User-Agent:** a continuación indicamos el nombre del bot al que vamos a permitir, o no fisgonear en nuestra web. Si estuviesemos pensando en permitir a los robots campar a sus anchas por nuestra web, evidentemente, no estariamos hablando de crear un archivo robots txt, con esta premisa, de los dos comandos directos **Allow** y **Disallow** el que vamos a usar es el disallow, ya que a los no indicados les vamos a dejar ver, hasta cierto punto, nuestros contenidos.

Si observamos el robots txt por defecto en Wordpress, nos pinta lo siguiente:

User-agent: *

Disallow: /wp-admin/

Qué nos dice este archivo, por un lugado que la instrucción es para todos los bots usando el asterisco *, y en segundo lugar que no les permita ver la carpeta wp-admin, hasta aquí correcto, pero como hemos visto antes, esta instrucción no nos sirve, ni para evitar el contenido duplicado ni para bloquear a los bots malignos, así que vamos un poco más allá:

Después de no pocas pruebas he llegado a la conclusión de que es totalmente insuficiente, ya que sigue dejando al descubierto muchas zonas de la web que no nos interesa para nada que estén visibles. Por ejemplo algo muy frecuente en estos últimos meses es encontrar multitud de intentos de ingreso en tu panel de administración usando bots que tratan de adivinar tu usuario y contraseña, por lo tanto un archivo al que aplicar el disallow sería wp-login, la secuencia quedaría así:

disallow: /wp-login

A partir de esta instrucción ningún bot puede acceder al archivo de login.

Vamos unos cuantos pasos más allá en la configuración avanzada de robots txt:

Ya hemos visto antes que el asterisco * sirve para conceder o negar a todos el acceso a distintas carpetas de nuestra web, pero existe otro símbolo muy importante, el símbolo de dollar $, vamos a ver como se utiliza y para que sirve. Imaginemos que en nuestra web, tenemos una url del tipo: http://eldominioquesea.com/novedades/ si aplicamos el disallow: /novedades/ lo que va a hacer inmediatamente es bloquear el acceso a esa página y todo aquello que va por detrás, es decir si tenemos un post bajo esa url /novedades/mi-post-mejor-escrito va a pasar desapercibido por las arañas y nunca va a ser indexado, bien, para decirle al robot que si puede acceder al contenido que va por debajo de esa página lo que haremos es usar el símbolo del dollar $, de esa forma el robot identifica que no queremos que indexe esa página concreta pero si aquello que hay detras de esa página y es contenido indexable, el comando quedaría así:

User-agent: *

Disallow: /novedades/$

Esto no es suficiente, ya que el operador asterisco también nos sirve para determinados casos y es mejor combinarlos.

Ya hemos visto que el dólar sirve para decirle que ahí termina la URL, que no puede llevar nada más por detrás todo aquello que deseemos aplicarle el “allow” o el “disallow”.
En el caso del asterisco le estamos diciendo que puede sustituir esa parte por lo que quiera, siempre que vaya seguido de “.htm”. Es decir, puede haber varios niveles de carpeta por medio (por ejemplo “/*carpeta*/*subcarpeta*/*pagina.htm*” también sería excluído).

Si analizamos el ejemplo le estamos diciendo a todos los robots que no indexen ningún archivo .**HTM** si bien le vamos a permitir, a través del dólar, que index, por ejemplo, todos los archivos con extensión .**HTML**.

Vamos a por el contenido duplicado y como evitarlo con robots txt.

El CMS de Wordpress genera algunas url, la de author, la de paginación, etc, que no queremos que sean indexadas para que no se interpreten como contenido duplicado.

Por ejemplo: http://eldominioquesea.com/index.php?page=2 que obviamente no queremos indexar y siguiendo con lo aprendido hasta ahora podriamos usar un disallow específico para esas terminaciones:

Disallow: /*?

Con esta línea de comando le decimos que no indexe nada que lleve "loquesea" (con el asterisco)pero siempre y cuando lleve una interrogación "?". No caigas en la tentación de simplemente poner un disallow: /*?*" para asegurarte en el caso de que después de la interrogación, si hay algo más tampoco lo indexe. Grave error, ya que el segundo asterisco otorga permiso para que se indexe todo aquello que va detrás de la interrogación.

Si has entendido bien hasta aquí, ahora deberías estar preguntando que al usar

User-agent: *

Disallow: /novedades

estamos negando el acceso a la carpeta completa, y por tanto a todo aquello que va detras, pero no es eso exactamente, ya que el bot va a entender que detras de esa expresión puede ir cualquier cosa, por tanto para excluir solo la página sin sus extensiones deberemos usar:

User-agent: *

Disallow: /novedades$

Otra pregunta que seguramente te haces a estas alturas es si se puede combinar los comandos Allow y los Disallow, sería lo ideal para perfeccionar el archivo robots txt la combinación de ambos, por ejemplo:

User-agent: *

Allow: /novedades/$

Disallow: /novedades/

Con esta sencilla instrucción le estamos diciendo al robot que SI indexe la página general de novedades, pero que no indexe las páginas siguientes con las novedades, recuerda respetar el orden, es mejor usar siempre el comando Allow y a continuación el Disallow, no todos los bots están igual de avanzados y les facilitamos un poco el trabajo.

Configurando el archivo robots txt para Wordpress:

Si una ventaja tiene Wordpress es que todas sus instalaciones son iguales, por lo cual hacer un archivo que nos valga para casi cualquier diseño web es fácil, después ya podrás entrar en que quieres que tengan acceso los robots y en que no quieres que fisgoneen los robots.

El archivo que te muestro a continuación es básico, ya que la utilización de plugins, personalización de temas y aplicaciones de terceros tendrás que revisar esta configuración básica:

La utilización de este formato de archivo es totalmente bajo tu responsabilidad, si algo no funciona revisa lo escrito hasta aquí.

User-agent: *

Disallow: /wp-login

Disallow: /wp-admin

Disallow: //wp-includes/

Disallow: /*/feed/

Disallow: /*/trackback/

Disallow: /*/attachment/

Disallow: /author/

Disallow: /*/page/

Disallow: /*/feed/

Disallow: /tag/*/page/

Disallow: /tag/*/feed/

Disallow: /page/

Disallow: /comments/

Disallow: /xmlrpc.php

Disallow: /*?s=

Disallow: /*/*/*/feed.xml

Disallow: /?attachment_id*

Bien, con este ejemplo ya evitamos la posible indexación de todas aquellas carpetas de sistema y archivos con sus correspondientes extensiones.

Pero siempre se puede ir un poco más allá y mejorar el robots txt.

Ya tenemos nuestro archivo básico de robots txt, pero tenemos más herramientas a nuestra disposición para indicar a los robots que deben hacer con determinadas páginas usando meta-etiquetas "robots"

Follow / NoFollow

Index / NoIndex

Index - Follow, permite la indexación y rastreo de la página, es la que encontrarás por defecto normalmente.

NoIndex - Follow, En este caso estamos indicando que no queremos que se indexe una determinada página pero si queremos que se rastree.

Index - NoFollow, Personalmente sólo utilizo este formato en post de terceros que contienen enlaces que no quiero que se rastreen, permito que la página se indexe pero no que se rastree.

NoIndex - NoFollow, salvo que tengas tu site en construcción no le veo utilidad a esta combinación, si no quieres indexar ni rastrear, manda la página a la papelera.

Combinando el Sitemap con robots txt:

Personalmente recomiendo incluir en robots txt la ruta al sitemap, así le ponemos fácil a los robots "amigos" encontrar el contenido de relevancia de nuestra web, basta con añadir (yo lo pongo al final) una línea parecida a esta:

Sitemap: http://eldominioquesea.com/sitemap.xml

Verifica tu url, ya que determinados plugins seo usan otras anatomías para nombrar el sitemap.

De todas formas recuerda que el robot de Google, va a indexar todo aquello que encuentre, está o no en el Sitemap, por lo cual la combinación con robots txt es vital, todo lo que queramos ocultar lo he explicado antes.

Bueno, y llegados hasta aquí, seguramente te preguntarás como podría ser un archivo robots txt infalible, pues te adjunto el que más utilizo, siempre con las variaciones asociadas a cada **diseño web** y en función de las directrices de **nuestro equipo de posicionamiento SEO.**

```
User-agent: *
Disallow: /wp-login
Disallow: /wp-admin
Disallow: //wp-includes/
Disallow: /*/feed/
Disallow: /*/trackback/
Disallow: /*/attachment/
Disallow: /author/
Disallow: /*/page/
Disallow: /*/feed/
Disallow: /tag/*/page/
Disallow: /tag/*/feed/
Disallow: /page/
Disallow: /comments/
Disallow: /xmlrpc.php
Disallow: /*?s=
Disallow: /*/*/*/feed.xml
Disallow: /?attachment_id*
User-agent: Orthogaffe
Disallow: /
# los rastreadores tendrían que ser amables y obedecer
# a menos que estén alimentando los motores de búsqueda.
User-agent: UbiCrawler
Disallow: /
User-agent: DOC
Disallow: /
User-agent: Zao
Disallow: /
User-agent: Twiceler
Disallow: /
# Algunos robots son conocidos por ser un problema, sobre todo los
destinadas a copiar
# sitios enteros o descargarlas para verlos sin conexión. Por favor
obedeced mi robots.txt.
#
User-agent: sitecheck.internetseer.com
Disallow: /
User-agent: Zealbot
Disallow: /
User-agent: MSIECrawler
Disallow: /
User-agent: SiteSnagger
Disallow: /
User-agent: WebStripper
Disallow: /
User-agent: WebCopier
Disallow: /
User-agent: Fetch
Disallow: /
User-agent: Offline Explorer
Disallow: /
User-agent: Teleport
Disallow: /

User-agent: TeleportPro
Disallow: /
User-agent: WebZIP
Disallow: /
User-agent: linko
```

```
Disallow: /
User-agent: HTTrack
Disallow: /
User-agent: Microsoft.URL.Control
Disallow: /
User-agent: Xenu
Disallow: /
User-agent: larbin
Disallow: /
User-agent: libwww
Disallow: /
User-agent: ZyBORG
Disallow: /
User-agent: Download Ninja
Disallow: /
User-agent: Nutch
Disallow: /
User-agent: spock
Disallow: /
User-agent: OmniExplorer_Bot
Disallow: /
User-agent: TurnitinBot
Disallow: /
User-agent: BecomeBot
Disallow: /
User-agent: genieBot
Disallow: /
User-agent: dotbot
Disallow: /
User-agent: MLBot
Disallow: /
User-agent: 80bot
Disallow: /
User-agent: Linguee Bot
Disallow: /
User-agent: aiHitBot
Disallow: /
User-agent: Exabot
Disallow: /
User-agent: SBIder/Nutch
Disallow: /
User-agent: Jyxobot
Disallow: /
User-agent: mAgent
Disallow: /
User-agent: MJ12bot
Disallow: /
User-agent: Speedy Spider
Disallow: /
User-agent: ShopWiki
Disallow: /
User-agent: Huasai
Disallow: /
User-agent: DataCha0s
Disallow: /
User-agent: Baiduspider
Disallow: /

User-agent: Atomic_Email_Hunter
Disallow: /
User-agent: Mp3Bot
Disallow: /
User-agent: WinHttp
```

```
Disallow: /
User-agent: betaBot
Disallow: /
User-agent: core-project
Disallow: /
User-agent: panscient.com
Disallow: /
User-agent: Java
Disallow: /
User-agent: libwww-perl
Disallow: /
# Francamente, wget en su modo recursivo es un problema frecuente.
# por tanto y para evitar sobrecargas
#
User-agent: wget
Disallow: /
#
#Este descarga millones de páginas sin ningún beneficio público
# http://www.webreaper.net/
User-agent: WebReaper
Disallow: /
# A continuación los bots desobedientes que no quieren
# hacer caso de robots.txt pero...
#
# el bot 'grub' es el más maleducado de todos
User-agent: grub-client
Disallow: /
User-agent: k2spider
Disallow: /
# Este manda tantos intentos por segundo que es molesto de narices
# http://www.nameprotect.com/botinfo.html
User-agent: NPBot
Disallow: /
```

#y no olvides tu Sitemap

Sitemap: http://eldominioquesea.com/sitemap.xml

Con este último ejemplo, además de optimizar nuestro archivo robots txt hemos dejado a los malos fuera de nuestra web, pero ello no significa que estés a salvo para siempre.