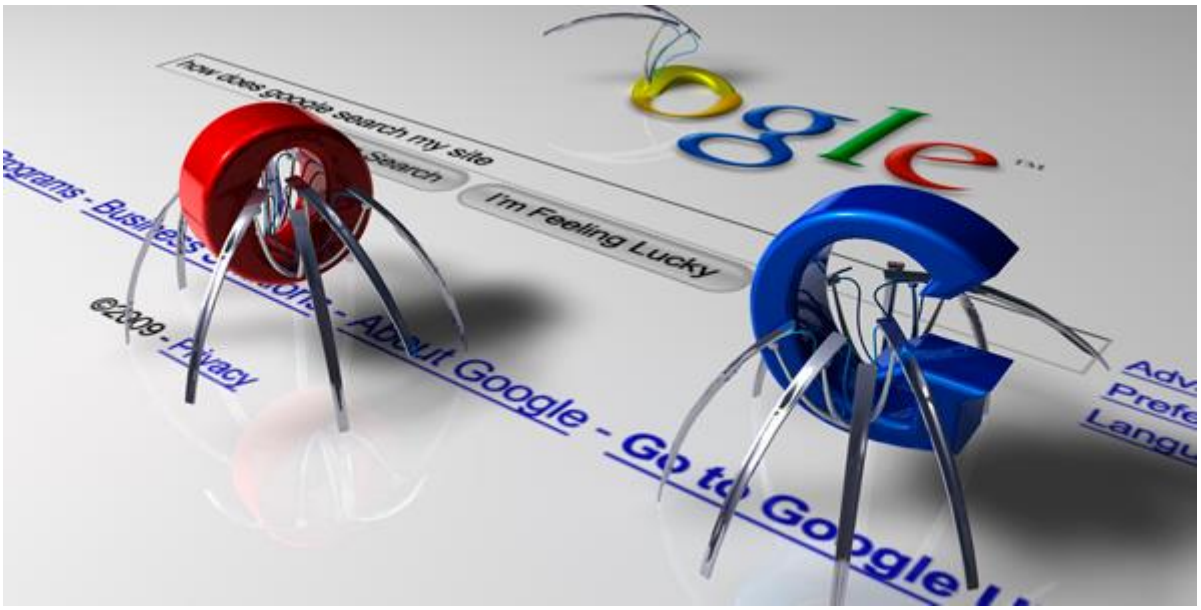
Robots.txt

Cómo ocultar partes de tu web a los buscadores



Cardenal Gardoki, 1
48008 BILBAO (Vizcaya)
Teléfono: 902 012 199
www.hostalia.com

HOSTALIA .com

Cuando un particular o una empresa desarrollan un portal web, su principal objetivo no es otro que llegar al mayor número de usuarios posibles. Para ello es muy importante conseguir un buen posicionamiento en los distintos buscadores que nos podemos encontrar en la actualidad.

Los programas encargados de revisar la información que hay en la red y de ir añadiendo contenido a los buscadores son los denominados robots, también conocidos con el nombre de arañas (spiders). Estos programas se dedican a rastrear las webs almacenando el contenido en sus bases de datos.

Estos robots realizan un análisis completo de nuestro sitio, pero hay ciertos casos en los que nos puede interesar que no se indexe un determinado archivo, carpeta o url. Para conseguir esto es necesario hacer uso de los denominados archivos robots.txt.

En esta página podéis encontrar un listado de todos los robots que hay actualmente funcionando.

¿Qué son los robots.txt?

Aunque el nombre del archivo pueda sugerir que se trata de algo muy complejo, un archivo robots.txt no es más que un texto plano que se coloca en el directorio raíz del sitio web y en el que nos podemos encontrar una serie de líneas de código con instrucciones para las arañas del contenido que no queremos que sea indexado. Todo lo que no se indique en este archivo será visitado e indexado por las arañas.

Cuando un robot llega a una web lo primero hace es buscar si en el directorio raíz se encuentra el archivo robots.txt. Si es así, el spider lo lee para conocer todas las restricciones que debe cumplir, y a continuación se pone a recorrer el sitio cumpliendo esas directrices.

Una mala configuración de este archivo nos puede hacer perder indexación en los buscadores, de ahí que sea de gran importancia saber cómo configurarlo de forma correcta, para indicar que únicamente no rastree aquello que no queremos que sea indexado.

Cabe decir que aunque de forma habitual los robots de los distintos buscadores suelen hacer caso a estas directivas, no significa que algunos de ellos las puedan obviar o interpretarlas de forma diferente. Además, también se pueden dar casos de que algún spammer desarrolle algún tipo de robot para conseguir direcciones de correo donde enviar correo basura que haga caso omiso a esas directrices.

Beneficios del uso de los archivos robots.txt

Un uso correcto de este tipo de archivos, puede reportarnos importantes beneficios. Vemos algunos de los más importantes.

- Denegar el acceso a nuestro sitio a determinados robots. Esto que puede parecer extraño no lo es si esos robots en vez de beneficiarnos lo que hacen es el efecto contrario.
- Mejorar el posicionamiento de nuestro sitio indicando a los spiders los sitios concretos que queremos indexar.
- Reducir la transferencia consumida en nuestro servidor, ya que al poder bloquear a ciertos robots o indicar aquellos sitios que no queremos que se indexen, estamos reduciendo el número de peticiones que se realiza a nuestro sitio.
- Impedir que se indexen archivos personales, archivos que pueden ser fotos, documentos, vídeos…
- Eliminar contenido duplicado. Con esto estamos impidiendo que los buscadores nos penalicen por encontrar en nuestro sitio distintas urls con la misma información.

Cómo funcionan los archivos robots.txt

```
User-agent: *
Disallow: /wp-admin
Disallow: /feed/
Disallow: /trackback/
Disallow: /comments/feed/
Disallow: /page/
Disallow: /comments/

Allow: /
Allow: /wp-contents/uploads/
```

A la hora de generar este tipo de archivos, debemos hacer uso de las directivas que nos proporcionan:

1.- User-agent

Se trata de una de las directivas más importantes y que nunca debe faltar a la hora de crear nuestro robots.txt. Por medio de esta directiva le estamos indicando para qué robots van orientadas las restricciones que indicaremos a continuación, ya que nos puede interesar que dependiendo del tipo de robot, pueda tener acceso a una u otras zonas. Su uso es el siguiente:

User-agent: nombre_robots

Por ejemplo:

User-agent: Googlebot

Si queremos que el bloqueo afecte a todos los robots, podemos utilizar el comodín asterisco (*).

User-agent: *

2.- Disallow

Se trata de la directiva que nos permite indicar aquellas carpetas o archivos que no se quieren indexar. Si queremos impedir que se indexen todos los archivos que forman parte de una carpeta, habría que poner al final del nombre la barra "/". Algunos ejemplos:

- **Disallow: /** Impediría la indexación de todo el sitio.
- **Disallow:** Permitiría la entrada a todos los directorios del sitio.
- **Disallow: /images/** Con esto estaríamos indicando que no accediera al directorio images.

Junto con esta directiva, también podemos hacer uso de ciertos comodines como son el asterisco (*) y el símbolo del dólar ($). El comodín * sirve para sustituir cualquier cadena, mientras que $ se utiliza para indicar que detrás no habrá nada más, sino que la ruta termina ahí.

Por ejemplo, si queremos impedir que sean indexados por cualquier robots aquellas imágenes que tengan extensión ".jpg", lo deberíamos indicar de la siguiente manera.

User-agent: *

Disallow: /*.jpg$

Otro ejemplo, supongamos que no queremos que sean indexadas por el robot MSNBot aquellas entradas en nuestro blog cuya url contenga el año 2010. En este caso, la regla sería la siguiente:

User-agent: MSNBot

Disallow: /2010/*

Tras el año 2010 ponemos el comodín * para indicar que puede ir cualquier tipo de cadena, pero no hacemos uso del $ porque no es el fin de la ruta, sino que detrás puede ir algo más como el mes en cuestión o el nombre de la entrada.

3.- Craw-delay

En muchas ocasiones, los robots bombardean nuestros sitios realizando cientos de peticiones para analizarlo. Estas peticiones pueden hacer que se llegue a colapsar nuestro portal. Para evitar esta situación,

podemos utilizar la directiva Craw-delay con la que indicamos al robot el tiempo que tiene que transcurrir entre uno y otro acceso. Veamos un ejemplo:

User-agent: Googlebot

Crawl-delay: 30

Con esto le estamos indicando al robot de Google que entre acceso y acceso deberían pasar 30 segundos.

4.- Visit-time

Se trata de una directiva que nos permite indicar a las arañas cuándo pueden revisar nuestro sitio. Por ejemplo, si queremos que sólo sea analizado de 4 de la mañana a 8 y media de la mañana, tendríamos que indicarlo de la siguiente forma.

Visit-time: 0400-0830

5.- Request-rate

Mediante esta directiva, lo que le indicamos al spider es el número de documentos que puede analizar cada cierto tiempo. Por ejemplo, si queremos que sólo analice un archivo cada 10 minutos, deberíamos indicarlo de la siguiente manera.

Request-rate: 1/10m

Las directivas que hemos visto anteriormente se pueden combinar entre si y aparecer tantas veces como queramos.

Por ejemplo. Supongamos que para el robot PHPDig no le vamos a dar acceso a nuestra carpeta de imágenes, pero sin embargo el resto de robots podrán acceder a todos los sitios excepto a las rutas que empiecen por la palabra tag, realizando el análisis de 3:30 am a 9 am, y con un retardo entre petición de 10 segundos.

User-agent: *

Disallow: /tag/*

Crawl-delay: 10

Visit-time: 0330-0900

User-agent: PHPDig

Disallow: /images/

Por ejemplo en el archivo Robots.txt de Hostalia tenemos los tags desindexados porque devuelven contenido duplicado, y le indicamos a las arañas dónde está el mapa del sitio para que lo indexen mejor:

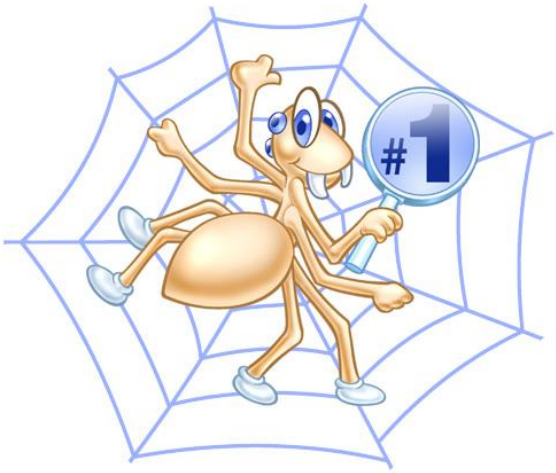
User-agent: *

Disallow: /tag/*

Disallow: /tag/

Sitemap: http://www.hostalia.com/sitemap.xml

Consejos para optimizar el archivo robots.txt



Tal es la importancia de este tipo de archivos que es fundamental contar con una buena optimización, ya que un error en ellos puede hacer que perdamos indexación en los buscadores de páginas que realmente sí que nos interesan posicionar.

Dependiendo del robot que se trate, cuando lee un archivo y detecta algún error en una directiva, puede actuar de dos formas:

- Ignora la que tiene el error y sigue leyendo el resto del archivo.
- Ignora todas las instrucciones que aparecen a partir de la que tiene el fallo de sintaxis.

Por este motivo es muy importante dedicar el tiempo necesario para optimizar nuestro robots.txt y evitar de esta forma cualquier tipo de error.

1.- Uso correcto de comodines

Los comodines son una herramienta fundamental para poder crear instrucciones más complejas que abarquen un gran número de urls que no queremos indexar. El problema es que no todos los navegadores las aceptan. Para evitar posibles problemas, lo mejor es que este tipo de directivas sean utilizadas al final del archivo una vez que hayamos definido todas las reglas para el resto de robots. De esta forma, nos aseguramos de que todos lo disallow anteriores sean respetados por los robots.

2.- Utilizar sólo etiquetas Disallow

Aunque no lo hemos comentado anteriormente, algunos robots también permiten el uso de etiquetas allow con las que indicar las rutas que sí deben ser indexadas. El uso de este tipo de etiquetas no tiene mucho sentido, ya que los robots asumen que si una url no aparece en una etiqueta disallow, significa que debe ser indexada.

3.- Uso de salto de línea

Para una mejor organización del archivo, es recomendable hacer uso de los saltos de línea entre los bloques que definen las reglas para un determinado rastreador. La estructura que se recomienda seguir es la de indicar primero el nombre de robot para el que van definida las reglas y a continuación las restricciones para ese robot en cuestión. A continuación se haría un salto de línea y se volvería a iniciar el siguiente bloque indicando el nombre del siguiente robot.

User-agent: nombre_robot_1

Disallow: /images/

User-agent: nombre_robot_2

Disallow: /videos/

4.- Mantener el archivo simple

Como ocurre en la mayoría de las ocasiones, cuanto más simple sea el archivo, más rápido será su ejecución por parte de los robots. Para ello, es recomendable no utilizarlos para bloquear el acceso a una url individual. Para ello podemos hacer uso de la etiqueta meta NOINDEX en la cabecera de esa página.

Cuanto más sencillo sea el archivo, menor probabilidad de que se cometan errores al crear el archivo y por lo tanto, mejor resultado podremos obtener del uso de este tipo de archivos.