

# Cómo Redirigir Una Página Web

En diferentes ocasiones un Webmaster necesita realizar una redirección, y afortunadamente existen diversas maneras de hacerlo. En este artículo te mostraremos como **redirigir una página web** y así evitar el tan [temido error 404](#). Para no tener problemas al redireccionar una URL, debemos de contar con la vieja dirección web, luego aplicar algún método para poder dirigir a la nueva URL.



Antes de comenzar, aprenderás un poco de HTTP ;)

## Códigos de respuesta HTTP

Cada vez que realizamos una petición de URL desde el navegador, el servidor envía además de datos, códigos de respuesta HTTP. Después de recibir la respuesta en el navegador, se pueden encontrar una respuesta como «hey estoy aquí, vamos!» (código de respuesta **HTTP 200 OK**). Existen diferentes tipos de código de respuesta HTTP, tal vez estés familiarizado con el 404; no solo las páginas web pueden responder con un estado 404, sino también cualquier otro archivo que le solicitemos al servidor, ya sea una imagen o cualquier otro tipo de activo.

Cada **respuesta HTTP** se clasifica bajo un cierto número de tres dígitos, por lo que 404 es un código de estado 4XX, y 200 está en la categoría 2XX. En nuestro caso particular, estamos interesados en la categoría 3XX de respuesta HTTP, como 301 o

302, lo que significa que ha sido movido permanentemente, porque estos son los códigos de estado específicamente reservados para las redirecciones.

En nuestro caso haremos un **redireccionamiento 301**, el cual nos sirve para avisarle a los navegadores o servidores proxy, que el sitio o pagina ha sido movida del sitio original a uno nuevo.

## Redirecciones HTML

Uno de los métodos mas sencillos para redirigir a otra dirección URL es con la **etiqueta Meta Refresh**. Podemos colocar esta etiqueta meta dentro de la etiqueta <head> en la parte superior de cualquier página HTML, justo como lo siguiente:

```
[html]&amp;lt;meta http-  
equiv=&amp;quot;refresh&amp;quot;  
content=&amp;quot;0;  
URL='http://nuevositioweb.com'&amp;quot;  
/&amp;gt;[/html]
```

El atributo content, es el tiempo que demora antes de que el navegador realice la redirección a la página nueva, con el ejemplo se redirige inmediatamente se cargue. Puedes ver que no es necesario establecer un código de estado HTTP. Aunque este método es la **manera más fácil para redirigir a una página web** hay algunas desventajas. Según el W3C hay algunos navegadores que no maneja la . Los usuarios pueden ver un flash como la página A se carga antes de ser redirigido a la página B. También deshabilita el botón Atrás de los navegadores más antiguos. No es una solución ideal, y se desaconseja utilizar en absoluto.

Una opción más segura sería redirigir el sitio web con JavaScript.

# Redirección HTML Con JavaScript

Llevar a una nueva URL con JavaScript es bastante fácil, simplemente tenemos que cambiar la ubicación de la propiedad en el objeto ventana:

```
[js>window.location =  
&amp;quot;http://nuevositio.com.co&quot;;[/js]
```

JavaScript es flexible, existen otras maneras de hacer esto:

```
[js]  
  
window.location =  
&amp;quot;http://nuevositio.com.co&quot;;  
window.location.href =  
&amp;quot;http://nuevositio.com.co&quot;;  
window.location.assign(&amp;quot;http://nuevositio.com.co&  
amp;quot;);  
window.location.replace(&amp;quot;http://nuevositio.com.co  
&amp;quot;);  
  
[/js]
```

Co este método existen unos cuantos problemas:

- Es necesario que JavaScript esté activado en el navegador para que funcione correctamente.
- Aún no está claro cómo los motores de búsqueda reaccionan a esta redirección.
- No hay códigos de respuesta de estado involucrados, por lo que no pueden depender de la información sobre la redirección.

Lo que necesitamos es una solución de servidor para ayudarnos mediante el envío de respuestas 301 a los buscadores y

navegadores.

# Redirección HTML Con Apache

Quizás el **método más común del redireccionamiento de una página web** es a través de la adición de reglas específicas en el `.htaccess` en un [servidor web Apache](#). De esta forma se puede tener un mayor control sobre el estado de las peticiones y lo que se desea hacer.

Para **redirigir a los usuarios a un nuevo sitio**, es necesario contar con un nuevo archivo `.htaccess` (o editar el existente) y añadimos al directorio raíz de la antigua web. Aquí está la regla necesaria para el redireccionamiento:

```
[bash]Redirect 301 / http://www.nuevositio.com[/bash]
```

Cualquier página que el usuario visita en el antiguo sitio web ahora serán dirigido a la nueva. Como se puede ver, ponemos el código de respuesta HTTP, justo después de la **regla para redirigir paginas en el htaccess**. Vale la pena mencionar que este tipo de redirección sólo funciona en [servidores Linux](#) con el **mod\_rewrite habilitado**, un módulo de Apache, que nos permite redirigir URL solicitadas en el servidor mediante la comprobación de un cierto patrón y, si se encuentra que el patrón, se modificará la solicitud de alguna manera.

Cuando es necesario **redireccionar todo el contenido**, hacia la URL nueva, en otras palabras **blog.com.co/post1** a **nuevoblog.com.co/post1**, es necesario hacer lo con la siguiente regla:

```
[bash]edirectMatch          301          /(.*)
http://www.nuevoblog.com.co$1[/bash]
```

O tal vez queremos redirigir las páginas individuales de forma específica. Para esto podemos añadir las reglas de la siguiente manera:

```
[bash]Redirect          301          /paginaanterior.html
http://www.sito/nuevapagina.html[/bash]
```

La redirección también es útil en casos de errores 404, los cuales no recomendables en un sitio web:

```
[bash]
&#amp;lt;IfModule mod_rewrite.c&#amp;gt;
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule .* 404.html [L]
&#amp;lt;/IfModule&#amp;gt;

[/bash]
```

Primero se debe comprobar si se tiene el **módulo mod\_rewrite activado**, si no es así, se debe activar.

# Redirección HTML Con Nginx

Si no usas Apache y en lugar usas **Nginx como servidor web**, a continuación, debes configurar el archivo nginx.conf, y agregar un bloque de servidor para manejar las solicitudes de redirección:

```
[bash]

server {
listen 80;
server_name old-website.com;
```

```
return 301 $scheme://new-website.com$request_uri;
}
```

```
[/bash]
```

Una vez más estamos usando la respuesta HTTP 301 y, con la variable de esquema, le solicitaremos http: // o https: // dependiendo del sitio web original utiliza.

## Redirección HTML Con Lighttpd

Para los servidores que ejecutan un servidor web con Lighttpd, puedes hacer una redirección HTML de la siguiente manera:

```
[bash]
```

```
server.modules = (
&&quot;mod_redirect&&quot;;
)

$HTTP[&&quot;host&&quot;] = ~
&&quot;^(www\.)?old-website.com&&quot; {
url.redirect = (
&&quot;^(.*)$&&quot; =&&gt;
&&quot;http://www.new-website.com/$1&&quot;;
)
}
```

```
[/bash]
```

## Redirección HTML Con PHP

Con PHP podemos utilizar la función header para la redirección, lo cual es bastante sencillo de usar:

```
[php]
```

```
&lt;?php
header('Location: http://www.nuevositioweb.com');
exit;
?>

[/php]
```

Esto tiene que ser puesto delante de cualquier tipo de código o contenido de cualquier otro tipo, sin embargo hay un pequeño problema. Por defecto, la función envía una respuesta 302 de redireccionamiento que le dice a todos que el contenido sólo se ha trasladado **temporalmente**. Teniendo en cuenta nuestro caso de uso específico que vamos a necesitar para mover de forma permanente los archivos a través de nuestra nueva página web, así que vamos a tener que hacer una redirección 301 en su lugar:

```
[php]

&lt;?php
header('Location: http://www.nuevositioweb.com/', true, 301);
exit();
?>

[/php]
```

El verdadero parámetro opcional anteriormente reemplazará un cabezazo ajustado previamente y el 301 al final es lo que cambia el código de respuesta para el correcto.

# Redirección HTML Con Node.js

Con [Node.js](#) también puedes realizar una redirección, para esto de debe **incluir el módulo http** y crear un nuevo servidor, seguido por el **método .writeHead ()**:

```
[js]
```

```
var http = require('http');  
  
http.createServer(function(req, res) {  
  res.writeHead(301, {Location: 'http://nuevositio.com'});  
  res.end();  
}).listen(8888);
```

```
[/js]
```

Si crear un nuevo archivo llamado `index.js` y pegas el código anterior y luego ejecutas `index.js`, en la línea de comandos puedes encontrar la versión local de la página web que te llevará a `http://nuevositio.com`. Pero para redirigir todos los artículos o paginas de la URL, tendremos que analizar la URL de la solicitud con el módulo `url` práctico de nodo:

```
[js]
```

```
var http = require('http');  
var url = require('url');  
  
http.createServer(function(req, res) {  
  var pathname = url.parse(req.url).pathname;  
  res.writeHead(301, {Location: 'http://nuevositio.com/' +  
  pathname});  
  res.end();  
}).listen(8888);
```

```
[/js]
```

Al usar la función `.writeHead()`, se puede añadir el nombre de ruta de la solicitud al final de la cadena URL. Ahora que va a redirigir a la misma ruta en nuestra nueva web.

Como puedes ver, redireccionar un sitio web no es para nada complicado, lo único que se debe tener en cuenta es el código de respuesta HTML para poder redireccionar de la forma correcta. Si conoces otra forma de redireccionar un sitio web,



de mejor manera que las anteriores, anímate y deja un comentario ;)