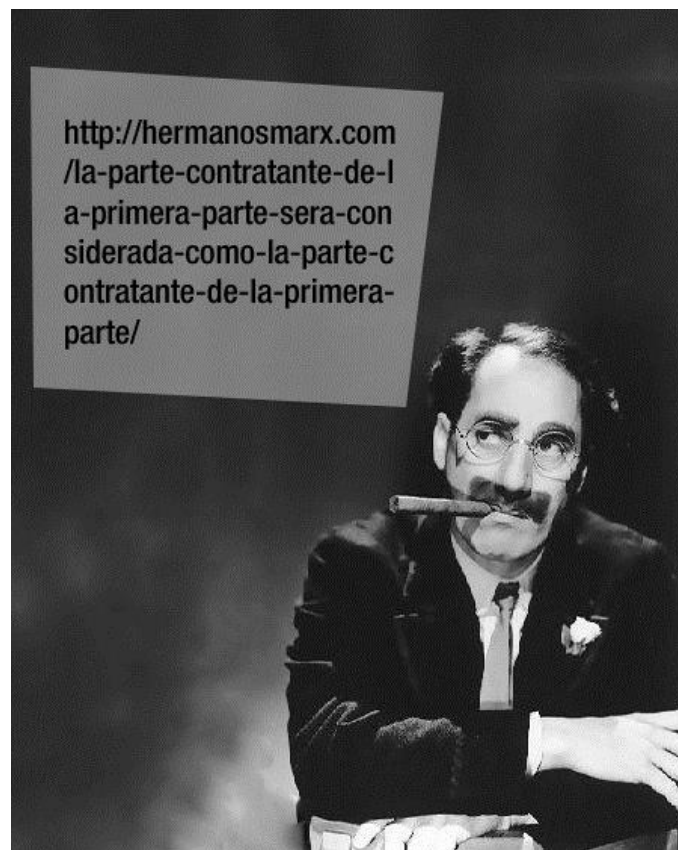


# La importancia de las URLs amigables



**acens**  
the *Cloud* hosting company

Calle San Rafael, 14  
28108 Alcobendas (Madrid)  
902 90 10 20  
[www.acens.com](http://www.acens.com)



## Introducción

Es habitual encontrarnos en Internet noticias sobre **posicionamiento web** donde se habla de la importancia del uso de **URLs amigables** (o friendly URL), un término muy utilizado pero que a muchos les puede sonar extraño.

De forma sencilla podemos decir que estas urls amigables **son direcciones web fáciles de recordar para el usuario** que visita un **sitio**, pero detrás de ellas hay mucha más información que no es detectada por el visitante y que sirve al servidor web para mostrar la información que ha sido solicitada, situación que se utiliza sobre todo cuando el contenido se genera de forma dinámica.

Por poner un ejemplo, estamos seguros que muchos de vosotros os habéis encontrado con direcciones web del siguiente estilo:

<http://www.miempresaweb.es/clientes.php?id=34>

Esa dirección está llamando al script “clientes.php” al que le pasa un “id”, para que haga una consulta a la base de datos y muestre la información correspondiente a ese usuario. Esta dirección vista de esa forma es complicada de recordar por los usuarios. Es lo que se conoce como “**url sucia**” o “dirty URL”.

Si en vez de encontrarla de esa forma, la vemos de esta otra la cosa cambia:

<http://www.miempresaweb.es/cliente/pedro>

Ahora el visitante en vez de ver el nombre del script que se ejecuta y los parámetros que se le pasan, ve nombres más legibles formando una url más sencilla de recordar. Además **así se describen mucho mejor los enlaces** dentro de una web y añade un mayor nivel de profesionalidad.

Por otro lado, las páginas estáticas están en formato HTML y también nos podemos encontrar casos en los que las rutas no sean amigables o no dicen nada al visitante. Por poner un ejemplo nos podríamos encontrar la siguiente dirección web para mostrar un formulario de contacto:

[http://www.miempresaweb.es/form\\_cont.html](http://www.miempresaweb.es/form_cont.html)

Hubiese sido mejor hacerlo de la siguiente forma que os mostramos, ya que se crea una dirección mucho más entendible para los usuarios.

<http://www.miempresaweb.es/contacto.html>

## ¿Qué nos aportan las URLs amigables?

El uso de urls amigables en nuestro sitio web nos proporciona unas interesantes ventajas:

- Crean **direcciones sencillas de recordar** y con información descriptiva del contenido para los visitantes.
- **Muestran la estructura del sitio**. Por ejemplo <http://www.miempresaweb.es/departamentos/contabilidad>
- Suelen tratarse de direcciones relativamente cortas que pueden ser transmitidas de forma verbal o por teléfono sin dar a equívocos.
- **Fáciles de escribir**, ya que este tipo de direcciones están formadas por palabras comunes, dejando de lado otros signos como interrogaciones, signo del igual...
- **Evitan los espacios en blanco** en los nombres de archivos y carpetas. Por ejemplo, un nombre de archivo como "características del hotel.html" crearía una URL como "características%20del%20hotel.html", cuando mediante el uso de url amigables podríamos obtener una url sin espacios en blanco como "características-del-hotel.html".
- Al estar **formadas con palabras claves**, los buscadores pueden llegar a interpretarlas y determinar su contenido, favoreciendo su **posicionamiento**.
- **Oculto la forma en la que está programada la página web** además de no mostrar los parámetros que les llegan. Esto impedirá mostrar información a posibles hackers que intenten entrar en nuestra web.
- Ofrece un aspecto más limpio y profesional, ayudando a mejora la estética del portal.

A pesar de todas los beneficios que nos pueden ofrecer este tipo de enlaces **hay algunas situaciones en las podríamos tener urls no amigables**, como pueden ser páginas donde el acceso sea restringido y que no son posicionadas por los buscadores, o en aquellas direcciones web donde se haga uso de un elevado número de parámetros que complique mucho la creación de una url amigable relativamente corta.

## Consejos a la hora de generar los textos que forman parte de las urls amigables

### First

<http://www.yoursite.com/referencia.php?id=34573>



### After

<http://www.yoursite.com/name/>

Como ocurre con la mayoría de las cosas en esta vida, no todo resultado es igual de bueno y no todas las urls amigables ofrecen el mismo resultado. Por ejemplo, anteriormente hemos hablado de que este tipo de direcciones permiten que sean recordadas por los usuarios, pero si esta es extremadamente larga, no obtendremos ese resultado esperado. Por eso es interesante tener en cuenta una serie de pautas a la hora de crear nuestras friendly urls. Veamos algunos de estos consejos para tener direcciones más cortas y mejor vistas por los buscadores:

- Evitar realizar direcciones web extremadamente largas y complejas.
- **Haz uso de palabras claves** por las que quieres posicionarte en los buscadores pero sin abusar de ellas, ya que podemos llegar a lograr el objetivo contrario.
- No utilizar a la hora de crear el enlace espacios en blanco ya que estos se suelen sustituir por los signos "%20". En su lugar es recomendable **usar los guiones medios (-)**.
- **No es recomendable hacer uso de los guiones bajos (\_)** ya que estos suelen coincidir con el subrayado de los enlaces y puede provocar equívocos.
- **Simplifica la dirección al máximo**, dejando de lado la mezcla de letras mayúsculas y minúsculas, ya que esto puede crear problemas a intentar escribir la dirección.
- **No utilices palabras como artículos, preposiciones o cualquier otro elemento que aporte poco valor.**

## Cómo utilizar URLs amigables con Apache haciendo uso del archivo .htaccess y mod\_rewrite



Cuando uno realiza una página dinámica usando un lenguaje de programación al lado del servidor como PHP o ASP, lo normal es que las urls que se formen no sean amigables, pero por suerte los servidores web ofrecen herramientas que ayudan a transformar las urls bonitas en direcciones entendibles para el servidor.

A continuación veremos cómo se puede hacer esto en un servidor web Apache mediante el uso del archivo **.htaccess** y el módulo de Apache **mod\_rewrite**, un módulo que permite manipular las URLs solicitadas.

Lo primero que tenemos que hacer es comprobar si este módulo está activo en nuestro [plan de alojamiento](#). Por norma general es así, pero para asegurarnos lo podemos hacer mediante el uso de la función “**phpinfo()**” de PHP. Para utilizarlo sólo debemos crear un fichero de extensión .PHP y dentro de él poner el siguiente código:

```
<?php phpinfo(); ?>
```

Esa instrucción nos mostrará la configuración de PHP en el servidor así como los módulos instalados. Deberíamos ver una imagen como la siguiente. Si mod\_rewrite está cargado, deberá aparecer indicado en la misma zona que os hemos resaltado en la imagen:

### PHP Credits

#### Configuration

#### apache2handler

Apache Version	Apache/2.2.21 (Win64) PHP/5.3.10
Apache API Version	20051115
Server Administrator	admin@localhost
Hostname:Port	localhost:80
Max Requests	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
Timeouts	Connection: 300 - Keep-Alive: 5
Virtual Server	No
Server Root	C:/wamp/bin/apache/Apache2.2.21
Loaded Modules	core mod_win32 mpm_winnt http_core mod_so mod_actions mod_alias mod_asis mod_auth_basic mod_authn_default mod_authn_file mod_authz_default mod_authz_groupfile mod_authz_host mod_authz_user mod_autoindex mod_cgi mod_dir mod_env mod_include mod_isapi mod_log_config mod_mime mod_negotiation <b>mod_rewrite</b> mod_setenvif mod_php5

Si lo tenemos activado, lo primero que debemos hacer es indicar en nuestro fichero .htaccess que vamos a hacer uso de la rescritura de urls. Para ello la primera línea que debemos poner en este archivo sería la siguiente:

### RewriteEngine On

Veamos a continuación un ejemplo sencillo de su uso. Supongamos que tenemos un archivo index.php con el siguiente enlace:

```
<a href="departamentos/administracion">Administración</a>
```

Como podemos ver, el enlace está formado por una url amigable y legible, pero nuestro **servidor web** no lo entiende. Para que nos muestre esa información hay que transformar la dirección en código entendible para el servidor. Debemos escribir en nuestro fichero .htaccess la expresión regular de rescritura (RewriteRule) capaz de interpretar esa url. En nuestro ejemplo deberíamos usar la siguiente línea:

```
RewriteRule ^departamentos/(\w+)/?$ departamentos.php?id=$1
```

Si no estás muy familiarizado con las **expresiones regulares** te recomendamos que visites el enlace anterior. De todas formas vamos a explicar brevemente qué significa cada una de las partes que forman la instrucción anterior.

- Evitar realizar direcciones web extremadamente largas y complejas.
  - **Haz uso de palabras claves** por las que quieres posicionarte en los buscadores pero sin abusar de ellas, ya que podemos llegar a lograr el objetivo contrario.
- 
- **RewriteRule.** Es la instrucción utilizada para indicar qué se hará una reescritura de url.
  - **^departamentos/(\w+)/?\$.** En este caso nos encontramos:
    - **^:** Para indicar el comienzo de la expresión regular.
    - **departamentos/:** Se le indica que la url asociada comienza por esa cadena de texto.
    - **(\w+):** Con esto indicamos que a continuación hay una palabra alfanumérica formada por uno o más caracteres. El valor lo almacena en \$1, en nuestro caso se almacenaría "administracion".
    - **/?:** Barra invertida opcional al final de la url.
    - **\$:** Para indicar el final de la expresión regular.
  - **departamentos.php?id=\$1.** En este caso nos encontramos:
    - **departamentos.php?id=:** Hace referencia al código que se ejecutará para la recuperación de la información.
    - **\$1:** variable donde se almacenará el valor llegado por la url.

En el ejemplo anterior sólo hemos utilizado un parámetro, que era el nombre del departamento, pero podemos incluir más de uno si nos hiciera falta.