

Redireccionamientos con Apache

En Apache, existen varios métodos para hacer redireccionamientos, que van desde las directivas simples hasta configuraciones avanzadas con `mod_rewrite`.

A continuación, se comentan las principales formas de hacerlo.

1. Usando las Directivas `Redirect` y `RedirectMatch`

Estas directivas son las más básicas.

- **Redirect:** redirige una URL completa o un directorio. Se usa en el archivo de configuración principal de Apache (`httpd.conf`) o en un archivo `.htaccess`.

```
Redirect 301 /antiguo-recurso http://ejemplo.es/nuevo-recurso
```

Esto redirige cualquier solicitud de `/antiguo-recurso` al nuevo recurso en `http://ejemplo.es/nuevo-recurso` con un código de estado 301 (permanente).

- **RedirectMatch:** permite utilizar expresiones regulares para crear redirecciones basadas en patrones.

```
RedirectMatch 301 "^/blog/([0-9]+)$" "http://ejemplo.es/post_$1"
```

Esto redirige cualquier URL que coincida con `/blog/123` a `http://ejemplo.es/post_123`.

2. Usando `mod_rewrite` para redirecciones avanzadas

El módulo `mod_rewrite` de Apache está pensado para redirecciones y reescrituras de URL más complejas. Por defecto no está habilitado en Apache por lo que hay que asegurarse de que lo está antes de empezar a utilizarlo.

- **Redirección Básica con `RewriteRule`:**

```
RewriteEngine On
```

```
RewriteRule ^antiguo-recurso$ /nuevo-recurso [R=301,L]
```

Esta regla redirige `http://ejemplo.es/antiguo-recurso` a `http://ejemplo.es/nuevo-recurso` usando una redirección 301.

- **Redirecciones Condicionales con `RewriteCond`:** Permite aplicar reglas basadas en condiciones específicas (por ejemplo, el protocolo, la IP del cliente, etc.).

```
RewriteCond %{HTTPS} off
```

```
RewriteRule ^(.*)$ https://%{HTTP_HOST}/$1 [R=301,L]
```

Esta configuración redirige todas las solicitudes HTTP a HTTPS.

3. Redirección por Nombre de Dominio con `VirtualHost`

Cuando se tienen varios dominios, se pueden hacer redirecciones usando bloques de `VirtualHost` en el archivo `httpd.conf` o en los archivos de configuración de sitios (`.htaccess`).

- **Redirección de un dominio completo:**

```
<VirtualHost *:80>
```

```
ServerName old-domain.com

Redirect permanent / http://new-domain.com/

</VirtualHost>
```

Esto redirige todo el tráfico de old-domain.com a new-domain.com de forma permanente.

4. Redirección a nivel de protocolo con Redirect y mod_rewrite

Para redirigir todo el tráfico HTTP a HTTPS, se puede usar Redirect en un VirtualHost o mod_rewrite.

- **Ejemplo con Redirect:**

```
<VirtualHost *:80>

    ServerName ejemplo.es

    Redirect / https://ejemplo.es/

</VirtualHost>
```

- **Ejemplo con mod_rewrite:**

```
RewriteEngine On

RewriteCond %{HTTPS} off

RewriteRule ^(.*)$ https://%{HTTP_HOST}/$1 [R=301,L]
```

5. Redirecciones permanente (301) o temporal (302)

Puedes especificar el tipo de redirección:

- **Redirección 301 (Permanente):** Se recomienda cuando el recurso ha cambiado de ubicación de forma definitiva. Ejemplo:
- Redirect 301 /pagina-antigua http://ejemplo.es/pagina-nueva
- **Redirección 302 (Temporal):** Se usa cuando el cambio es temporal, por ejemplo durante un mantenimiento.
- Redirect 302 /mantenimiento http://ejemplo.es/mantenimiento-temporal

6. Redirecciones con ErrorDocument

Aunque principalmente se usa para manejar errores personalizados, ErrorDocument también puede servir para redirigir URLs no encontradas (404) a una página específica.

```
ErrorDocument 404 /pagina-error.html
```

7. Cuándo utilizar mod_rewrite

El módulo mod_rewrite en Apache ofrece una flexibilidad y potencia mucho mayor que las directivas básicas Redirect y RedirectMatch para gestionar redirecciones y reescrituras de URL. A continuación se detallan las principales ventajas de mod_rewrite y qué aporta en comparación con las directivas estándar de redirección

7.1. Para aplicar condiciones avanzadas

- **mod_rewrite permite establecer condiciones** para las redirecciones, usando RewriteCond. Puedes aplicar reglas solo si ciertas condiciones son verdaderas, como verificar el tipo de navegador del usuario, el origen de la solicitud, la existencia de archivos o directorios, o incluso el valor de cookies y variables de entorno.
- Por ejemplo, podrías crear una redirección solo si el usuario está en un dispositivo móvil, algo que no es posible con Redirect o RedirectMatch:
- RewriteCond %{HTTP_USER_AGENT} "Mobile|Android|iPhone"
- RewriteRule ^\$ /mi_sitio/version_movil.html [L]

7.2. Para hacer reescritura interna sin redirección visible

- mod_rewrite permite reescribir URLs internamente, es decir, cambiar la ruta del recurso que Apache procesa sin que el usuario vea ningún cambio en la barra de direcciones. Esto es útil para crear URLs amigables (por ejemplo, convertir /producto/123 en /producto.php?id=123 sin cambiar la URL visible).
- Con Redirect y RedirectMatch, cada cambio en la URL es visible para el usuario, ya que estas directivas solo pueden redirigir (no reescribir internamente).
- RewriteRule ^producto/([0-9]+)\$ producto.php?id=\$1 [L]

7.3. Con expresiones regulares complejas y patrones dinámicos

- mod_rewrite permite un uso avanzado de **expresiones regulares** para definir reglas de reescritura y redirección. Esto permite crear reglas muy detalladas y específicas para URLs con patrones complejos, como múltiples parámetros opcionales, segmentos variables, etc.
- Si bien RedirectMatch también soporta expresiones regulares, mod_rewrite permite combinarlas con otras directivas y condiciones de una forma mucho más flexible.

7.4. Para controlar el comportamiento usando flags

- mod_rewrite tiene una amplia variedad de flags que modifican el comportamiento de cada regla, como:
 - L (Last): indica que si se cumple esta regla, no se deben procesar más reglas de reescritura.
 - R=301: especifica un código de estado HTTP, permitiendo redirecciones permanentes (301) o temporales (302).
 - QSA (Query String Append): permite añadir cadenas de consulta a las URLs reescritas.
 - NC (No Case): ignora mayúsculas y minúsculas al evaluar reglas.

Estos flags permiten un control granular sobre cómo se aplican y se comportan las redirecciones y reescrituras, lo cual no es posible con Redirect y RedirectMatch.

7.5. Redirecciones Condicionadas por Variables del Servidor

- mod_rewrite permite basar las reglas en **variables del servidor y del entorno**, como la dirección IP del usuario, el protocolo (HTTP o HTTPS), o el tipo de solicitud (GET, POST,

etc.). Por ejemplo, puedes redirigir a usuarios en HTTPS si intentan acceder a través de HTTP:

- RewriteCond %{HTTPS} off
- RewriteRule ^(.*)\$ https://%{HTTP_HOST}/\$1 [R=301,L]

7.6. Para crear de URLs "amigables"

- Con mod_rewrite, puedes fácilmente convertir URLs técnicas en URLs amigables que son más fáciles de leer y que pueden ayudar en el SEO (Search Engine Optimization). Por ejemplo, convertir una URL como /producto.php?id=123 en /producto/123.