



# **LABORATORIO SEO**

## **Cómo indexa Google elementos movidos con jQuery**

**WHITE PAPER 2**

<https://puntorojo.agency/>

## Cómo indexa Google elementos movidos con jQuery

jQuery es una biblioteca de JavaScript que permite manipular elementos del template de una página web con facilidad. Realizamos 13 experimentos utilizando jQuery con diferentes variables.

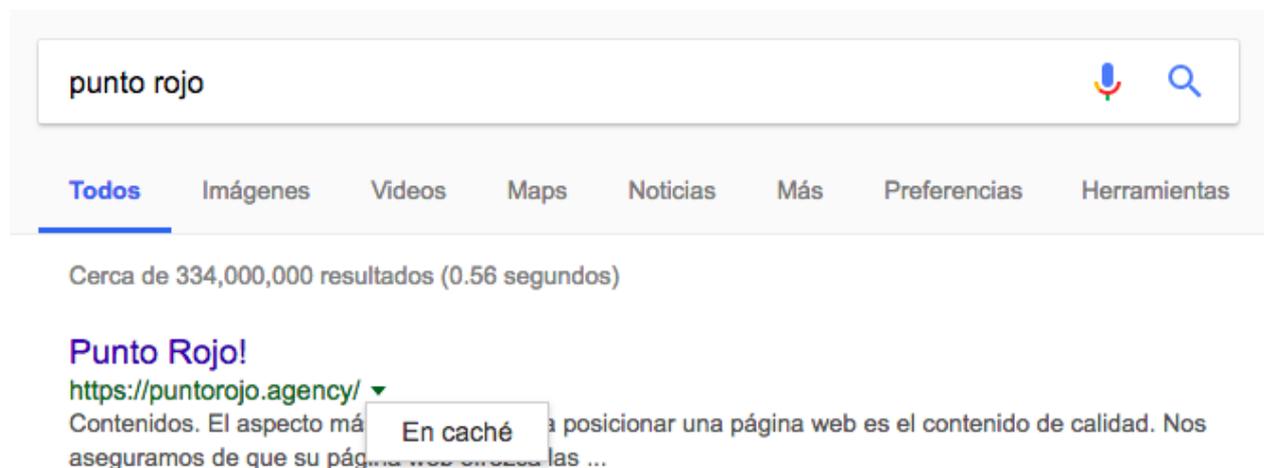
Mediante métodos implementados con jQuery, alteramos el orden de elementos HTML con el objetivo de conocer cómo se comporta Google al indexar estos contenidos.

Al agregar páginas a su índice, Google guarda una versión de las mismas en sus servidores. A estas páginas guardadas se les llama páginas en caché, y existen de dos tipos:

- **Versión completa:** versión completa de la página web.
- **Solo texto:** solo incluye texto.

Analizar las versiones que guarda Google en caché puede resultar importante para el posicionamiento, ya que nos permite ver qué entiende Google y qué no.

Para ver la versión en caché de una página, podemos buscarla en Google y hacer clic en la flecha que aparece en el resultado al lado de la URL. Se desplegará un botón para ver la página en caché.



A continuación, detallamos cómo Google interactuó con cada una de esas pruebas. **El objetivo es determinar si Google entiende este código, cómo lo comprende, si lo indexa y cuándo lo muestra en sus resultados de búsqueda.**

**Tipo de prueba:** Crawlabilidad e indexabilidad de elementos movidos con jQuery

**Objetivo:** Testear si Google interpreta e indexa elementos movidos con jQuery

**Forma de testeo:** Dentro de un código html, se mueve un elemento mediante el método jQuery 'prependTo'. El elemento destino se identifica con un id.

**Variable:** "Nicolas" esta en la posición 3 de la lista de nombres. Por medio de jQuery es movido a la posición 1.

### Código fuente de la página:

```
1 <html>
2 <head>
3   <script src="https://code.jquery.com/jquery-1.10.2.js"></script>
4 </head>
5 <body>
6
7 <ul id="TodosLosNombres">
8   <li id="nombre1">Alberto</li>
9   <li id="nombre2">Alfredo</li>
10  <li id="nombre3">Nicolas</li>
11  <li id="nombre4">Agustin</li>
12 </ul>
13
14
15 <script>
16 $( "#nombre3" ).prependTo( "#TodosLosNombres" );
17 </script>
18 </body>
19
20 </html>
21
```

**En la versión 'solo texto' del caché de Google, "Nicolás" sigue estando en posición 3:**



The screenshot shows a browser window displaying Google's cache of a webpage. The address bar shows the URL: `webcache.googleusercontent.com/search?q=cache:`. Below the address bar, there is a message: "This is Google's cache of [redacted] it is a snapshot of the page as it appeared on 29 May 2018 18:19:04 GMT." There are three links: "Full version", "Text-only version", and "View source". A tip says: "Tip: To quickly find your search term on this page, press Ctrl+F or ⌘-F (Mac) and use the find bar." Below this, there is a bulleted list of names: Alberto, Alfredo, Nicolas, and Agustin.

Sin embargo, tanto en la versión completa del caché de Google como en el snippet de resultados de búsqueda, **Nicolás se encuentra en el puesto 1.**



webcache.googleusercontent.com/search?q=cache:

This is Google's cache of [redacted] It is a snapshot of the page as it appeared on 29 May 2018 18:19:04 GMT.

[Full version](#) [Text-only version](#) [View source](#)

Tip: To quickly find your search term on this page, press **Ctrl+F** or **⌘-F** (Mac) and use the find bar.

- Nicolas
- Alberto
- Alfredo
- Agustin



site:[redacted] nicolas

[Todos](#) [Imágenes](#) [Maps](#) [Noticias](#) [Videos](#) [Más](#) [Preferencias](#) [Herramientas](#)

Cerca de 1 resultados (0.24 segundos)

Tst [redacted]  
www [redacted] [Traducir esta página](#)

## Conclusión:

- Se puede mover contenido para SEO haciendo uso del método "prependTo" localizando el destino por su id.

## jQuery - Prueba 2

**Tipo de prueba:** Crawleabilidad e indexabilidad de elementos movidos con jQuery

**Objetivo:** Testear si Google interpreta e indexa elementos movidos con jQuery

**Forma de testeo:** Dentro de un código html, se mueve un elemento mediante el método jQuery "insertAfter". El elemento destino se identifica con un id.

**Variable:** El h1 esta por debajo del <p> en el código html. Por medio de jQuery es movido encima del <p>.

### Código fuente de la página:

```

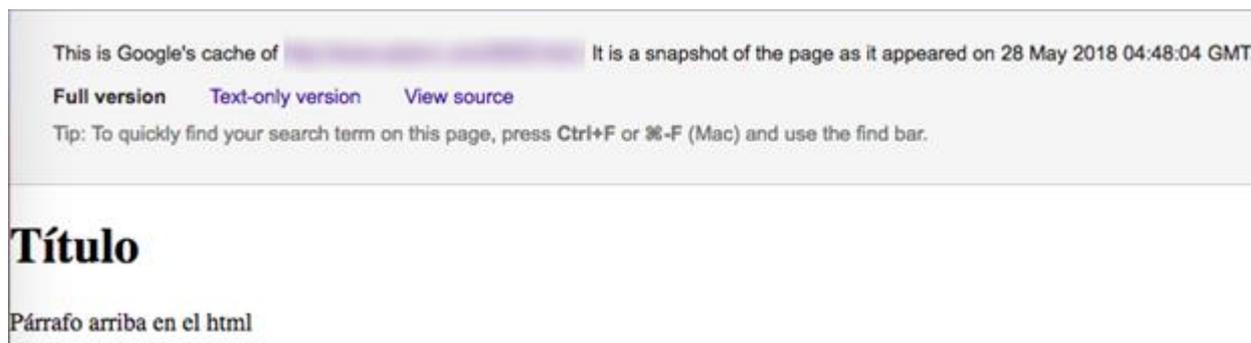
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
5 </head>
6 <body>
7 <p> Párrafo arriba en el html </p>
8 <h1 id="foo">Título</h1>
9
10 <script>
11 $( "p" ).insertAfter( "#foo" );
12 </script>
13
14 </body>
15 </html>

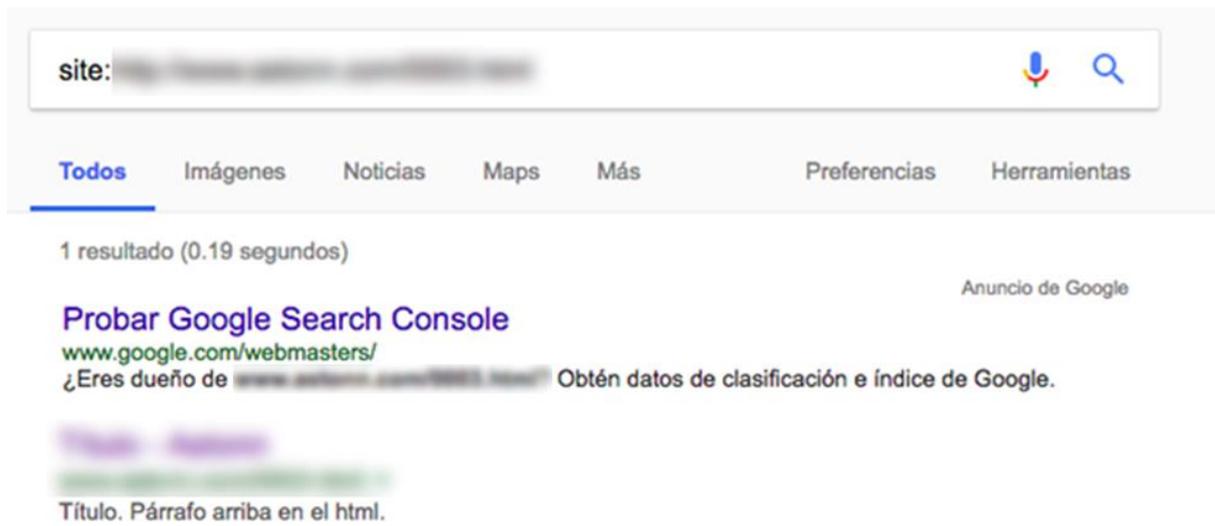
```

En la versión 'solo texto' del caché de Google, el h1 sigue estando debajo del <p>.



Sin embargo, tanto en la versión completa del caché de Google como en el snippet de resultados de búsqueda, **el título h1 se encuentra indexado en el puesto 1.**





**Conclusión:**

- Se puede mover contenido para SEO haciendo uso del método "insertAfter" de jQuery localizando el destino por su id.

## jQuery - Prueba 3

**Tipo de prueba:** Crawleabilidad e indexabilidad de elementos movidos con jQuery

**Objetivo:** Testear si Google interpreta e indexa elementos movidos con jQuery

**Forma de testeo:** Dentro de un código html, se mueve un elemento encima de otro, utilizando el método insertBefore de jQuery. El elemento de origen se identifica por su id.

**Variable:** El h1 está por debajo del <p> en el código html. Por medio de jQuery es movido encima del <p>.

**Código fuente de la página:**

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
5 </head>
6 <body>
7 <p id="parrafo1"> Al igual que en los casos anteriores, este Párrafo se encuentra más arriba que el título en el html </p>
8 <h1 id="titulo1">Título principal</h1>
9
10 <script>
11 $( "#titulo1" ).insertBefore( "#parrafo1" );
12 </script>
13
14 </body>
15 </html>

```

En la versión 'solo texto' del caché de Google, el h1 sigue estando debajo del <p>.



Sin embargo, tanto en la versión completa del caché de Google como en el snippet de resultados de búsqueda, el título h1 se encuentra indexado en el puesto 1.

← → ↻ ⓘ webcache.googleusercontent.com/search?q=cache

This is Google's cache of [http://www.punto rojo.com/2008/10/17/](#). It is a snapshot of the page as it appeared on 29 Mar 2008 17:10:36 GMT. The **current page** could have changed in the meantime. [Learn more.](#)

[Full version](#)   [Text-only version](#)   [View source](#)

Tip: To quickly find your search term on this page, press **Ctrl+F** or **⌘-F** (Mac) and use the find bar.

## Título principal

Al igual que en los casos anteriores, este Párrafo se encuentra más arriba que el título en el html

site:   

[Todos](#)   [Imágenes](#)   [Noticias](#)   [Maps](#)   [Más](#)   [Preferencias](#)   [Herramientas](#)

1 resultado (0.17 segundos) Anuncio de Google

### Probar Google Search Console

[www.google.com/webmasters/](http://www.google.com/webmasters/)  
¿Eres dueño de [www.punto rojo.com/2008/10/17/](#)? Obtén datos de clasificación e índice de Google.

**Conclusión:**

- Se puede mover contenido para SEO haciendo uso del método "insertBefore" de jQuery localizando el destino por su id.

## jQuery - Prueba 4

**Tipo de prueba:** Crawleabilidad e indexabilidad de elementos movidos con jQuery

**Objetivo:** Testear si Google interpreta e indexa elementos movidos con jQuery

**Forma de testeo:** Se cambia el orden de un elemento identificándolo por medio de su clase, con el método jQuery "insertAfter"

**Variable:** El <h2> está por debajo del <h1> en el código fuente. Es identificado por su ClassName y movido a primera posición mediante un insertAfter.

**Código fuente de la página:**

```

1 <html>
2 <head>
3   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
4 </head>
5 <body>
6   <div id="cajal">
7     </div>
8 <h1>Usuario 2</h1>
9 <h2 class="encabezado">Usuario 1</h2>
10
11 <script type="text/javascript" >
12 $( document.getElementsByClassName("encabezado")).insertAfter( "#cajal" );
13
14
15 </script>
16 </body>
17 </html>

```

En la versión 'solo texto' del caché de Google, el h2 sigue estando debajo del h1.

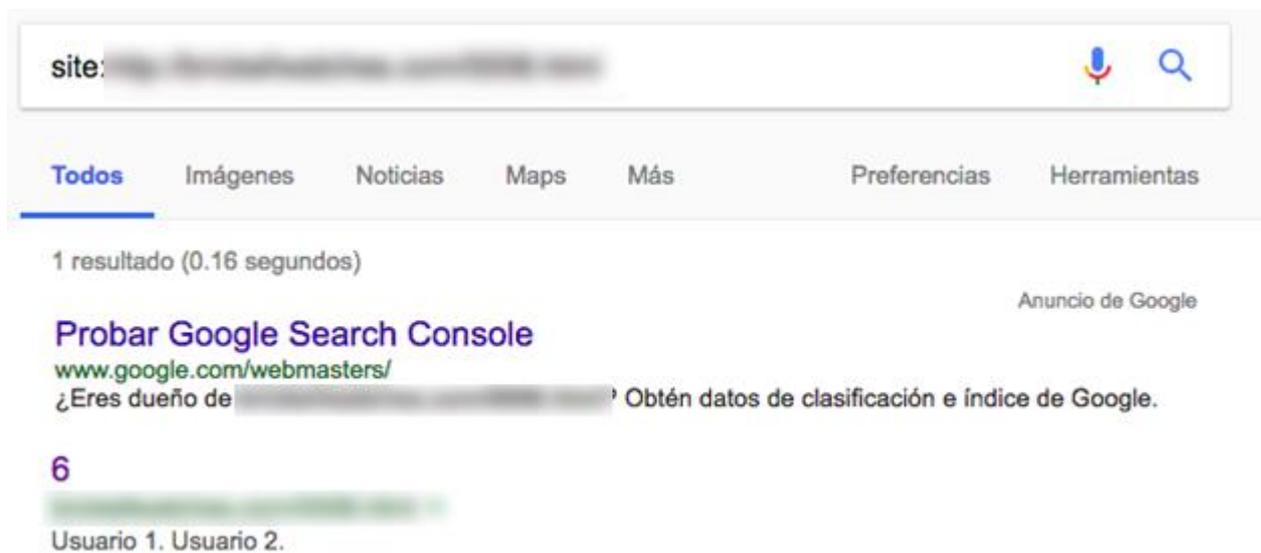


Sin embargo, tanto en la versión completa del caché de Google como en el snippet de resultados de búsqueda, el título h2 se encuentra indexado en el puesto 1.



**Usuario 1**

**Usuario 2**



**Conclusión:**

- Se puede mover contenido para SEO haciendo uso del método "insertAfter" de jQuery localizando el destino por su nombre de clase.

## jQuery - Prueba 5

**Tipo de prueba:** Crawleabilidad e indexabilidad de elementos movidos con jQuery

**Objetivo:** Testear si Google interpreta e indexa elementos movidos con jQuery

**Forma de testeo:** Se cambia el orden de un elemento con el método jQuery "insertAfter" identificándolo por medio de su id.

**Variable:** El h2 esta por debajo del <h1> en el código html. Por medio de jQuery es movido encima del <h1>.

**Código fuente de la página:**

```

1 <html>
2 <head>
3   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
4 </head>
5 <body>
6   <div id="cajal">
7
8     </div>
9 <h1>Usuario 2</h1>
10 <h2 id="encabezado">Usuario 1</h2>
11
12 <script type="text/javascript" >
13   $( document.getElementById("encabezado")).insertAfter( "#cajal" );
14
15
16 </script>
17 </body>
18 </html>

```

En la versión 'solo texto' del caché de Google, el h2 sigue estando debajo del h1.

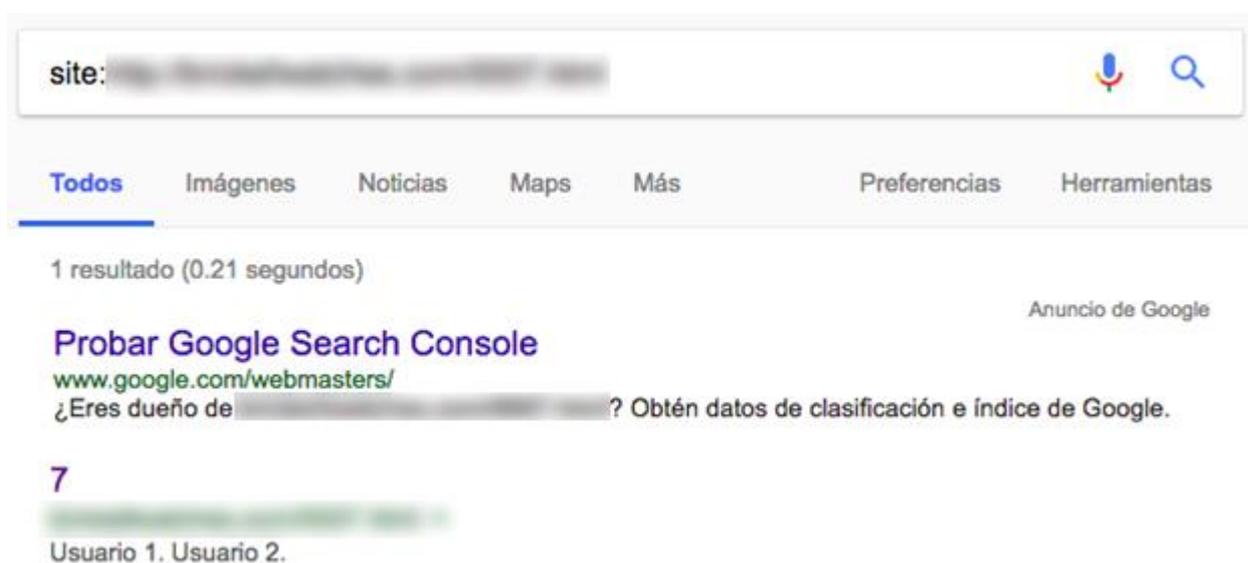


Sin embargo, tanto en la versión completa del caché de Google como en el snippet de resultados de búsqueda, el título h2 se encuentra indexado en el puesto 1.



**Usuario 1**

**Usuario 2**



**Conclusión:**

- Se puede mover contenido para SEO haciendo uso del método "insertAfter" de jQuery localizando el destino por su id.

## jQuery - Prueba 6

**Tipo de prueba:** Crawleabilidad e indexabilidad de elementos movidos con jQuery

**Objetivo:** Testear si Google interpreta e indexa elementos movidos con jQuery

**Forma de testeo:** Se cambia el orden de un elemento con el método jQuery "insertAfter" identificándolo por medio de su tagName.

**Variable:** El h2 esta por debajo del <h1> en el código html. Por medio de jQuery es movido encima del <h1>.

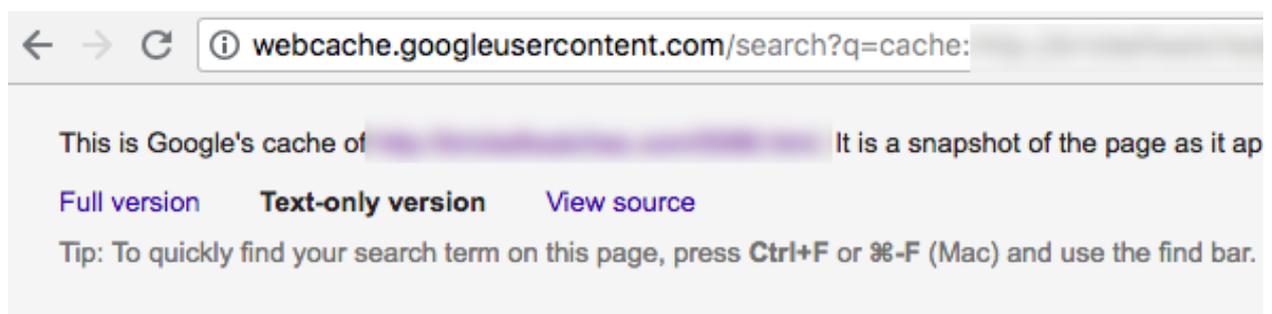
### Código fuente de la página:

```

1 <html>
2 <head>
3   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
4 </head>
5 <body>
6   <div id="cajal">
7
8     </div>
9 <h1>titulo 2</h1>
10 <h2>titulo 1</h2>
11
12 <script type="text/javascript" >
13 $( document.getElementsByTagName("h2")).insertAfter( "#cajal" );
14
15
16 </script>
17 </body>
18 </html>

```

En la versión 'solo texto' del caché de Google, el h2 sigue estando debajo del h1.



**titulo 2**

**titulo 1**

Sin embargo, tanto en la versión completa del caché de Google como en el snippet de resultados de búsqueda, el título h2 se encuentra indexado en el puesto 1.

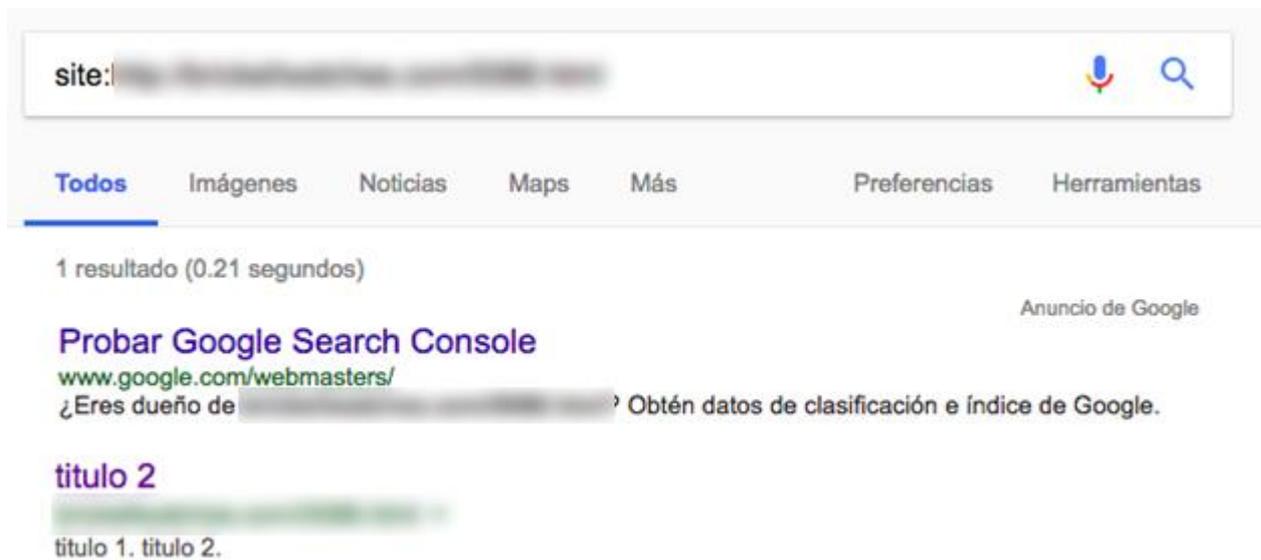
This is Google's cache of [\[URL\]](#) It is a snapshot

[Full version](#)   [Text-only version](#)   [View source](#)

Tip: To quickly find your search term on this page, press **Ctrl+F** or **⌘-F** (Mac).

## titulo 1

## titulo 2



site: [URL]

[Todos](#)   [Imágenes](#)   [Noticias](#)   [Maps](#)   [Más](#)   [Preferencias](#)   [Herramientas](#)

1 resultado (0.21 segundos)

Anuncio de Google

**Probar Google Search Console**  
[www.google.com/webmasters/](http://www.google.com/webmasters/)  
¿Eres dueño de [URL]? Obtén datos de clasificación e índice de Google.

**titulo 2**  
[URL]  
titulo 1. titulo 2.

### Conclusión:

- Se puede mover contenido para SEO haciendo uso del método "insertAfter" de jQuery localizando el destino por su nombre de etiqueta.

## jQuery - Prueba 7

**Tipo de prueba:** Crawleabilidad e indexabilidad de elementos movidos con jQuery

**Objetivo:** Testear si Google interpreta e indexa elementos movidos con jQuery

**Forma de testeo:** Se cambia el orden de un elemento con el método jQuery "insertBefore" identificándolo por medio de su TagName.

**Variable:** El h2 esta por debajo del <h1> en el código html. Por medio de jQuery es movido encima del <h1>.

**Código fuente de la página:**

```
1 <html>
2 <head>
3   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
4 </head>
5 <body>
6   <div id="cajal">
7
8     </div>
9 <h1>titulo 2</h1>
10 <h2>titulo 1</h2>
11
12 <script type="text/javascript" >
13 $( document.getElementsByTagName("h2")).insertBefore( "#cajal" );
14
15
16 </script>
17 </body>
18 </html>
```

En la versión 'solo texto' del caché de Google, el h2 sigue estando debajo del h1.

Esta es la memoria caché de Google de [\[URL\]](#) Es una instantánea de la página se tanto. [Más información.](#)

[Versión completa](#)   [Versión de sólo texto](#)   [Ver origen](#)

Consejo: para encontrar tu término de búsqueda rápido en esta página, presiona Ctrl+F o ⌘-F (Mac) y usa la bar

**titulo 2**

**titulo 1**

Sin embargo, tanto en la versión completa del caché de Google como en el snippet de resultados de búsqueda, el título h2 se encuentra indexado en el puesto 1.

Esta es la memoria caché de Google de [redacted] Es una instantánea de la página tanto. [Más información.](#)

[Versión completa](#)   [Versión de sólo texto](#)   [Ver origen](#)

Consejo: para encontrar tu término de búsqueda rápido en esta página, presiona **Ctrl+F** o **⌘-F** (Mac) y usa

## titulo 1

## titulo 2

site: [redacted]  

[Todos](#)   [Imágenes](#)   [Noticias](#)   [Maps](#)   [Más](#)   [Preferencias](#)   [Herramientas](#)

1 resultado (0.19 segundos) Anuncio de Google

**Probar Google Search Console**  
[www.google.com/webmasters/](http://www.google.com/webmasters/)  
¿Eres dueño de [redacted] Obtén datos de clasificación e índice de Google.

[redacted]  
[redacted]

titulo 1. titulo 2.

### Conclusión:

- Se puede mover contenido para SEO haciendo uso del método "insertBefore" de jQuery localizando el destino por su nombre de etiqueta.

## jQuery - Prueba 8

**Tipo de prueba:** Crawleabilidad e indexabilidad de elementos movidos con jQuery

**Objetivo:** Testear si Google interpreta e indexa elementos movidos con jQuery

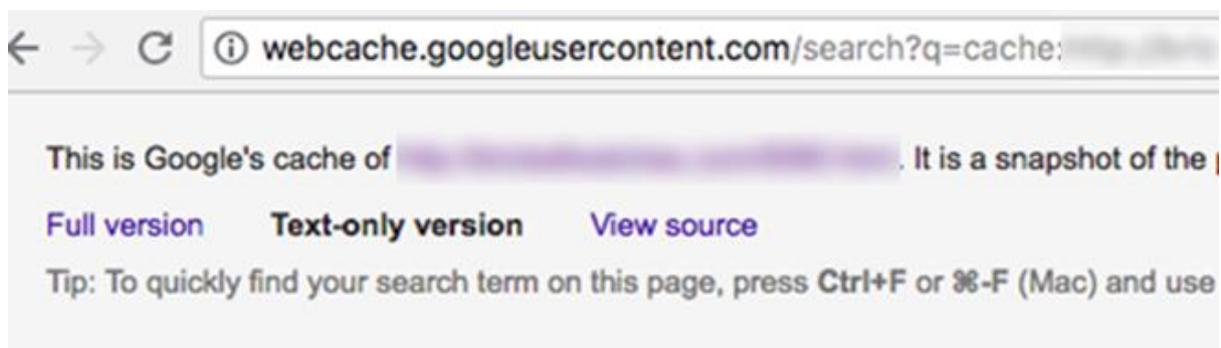
**Forma de testeo:** Se cambia el orden de un elemento con el método jQuery "prependTo" identificándolo por medio de su TagName.

**Variable:** "Ramiro" esta al final de la lista de nombres. Es identificado por medio de su tag name y movido al primer puesto por medio de jQuery.

### Código fuente de la página:

```
1 <html>
2 <head>
3     <script src="https://code.jquery.com/jquery-1.10.2.js"></script>
4 </head>
5 <body>
6
7 <ul id="names">
8     <li id="nombre1">Roberto</li>
9     <li id="nombre2">Ruffo</li>
10    <li id="nombre3">Delfina</li>
11    <p>Ramiro</p>
12 </ul>
13
14
15 <script>
16 $( document.getElementsByTagName( "p" ) ).prependTo( "#names" );
17 </script>
18 </body>
19
20 </html>
21
```

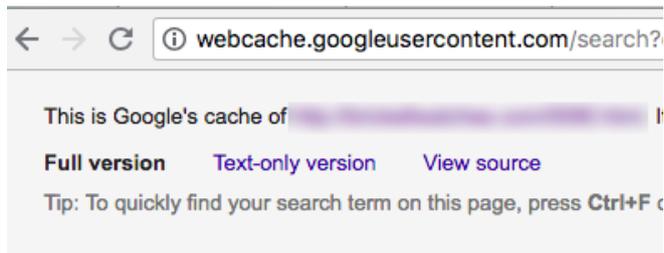
En la versión 'solo texto' del caché de Google, el elemento se encuentra al final de la lista.



- Roberto
- Ruffo
- Delfina

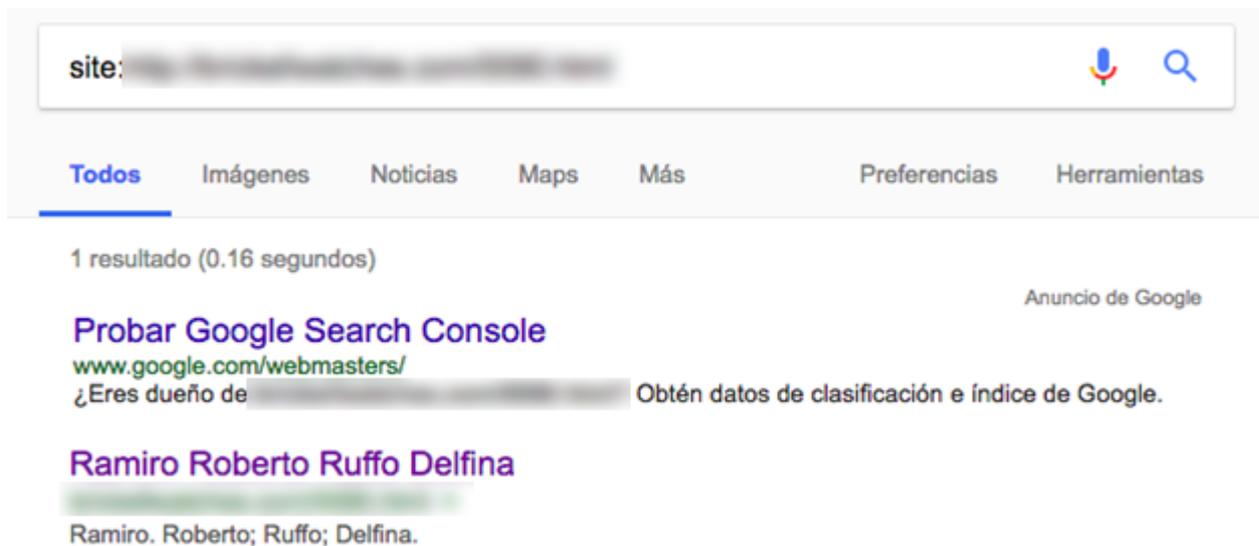
Ramiro

Sin embargo, tanto en la versión completa del caché de Google como en el snippet de resultados de búsqueda, el título elemento movido por jQuery se encuentra en primer lugar.



Ramiro

- Roberto
- Ruffo
- Delfina



## Conclusión:

- Se puede mover contenido para SEO haciendo uso del método "prependTo" de jQuery localizando el destino por su nombre de etiqueta.

## jQuery - Prueba 9

**Tipo de prueba:** Crawleabilidad e indexabilidad de elementos movidos con jQuery

**Objetivo:** Testear si Google interpreta e indexa elementos movidos con jQuery

**Forma de testeo:** Se cambia el orden de un elemento con el método jQuery "insertAfter" identificándolo por medio de su tagName. El jQuery se ejecuta desde un archivo .js externo.

**Variable:** El h2 esta por debajo del <h1> en el código html. Por medio de jQuery es movido encima del <h1>.

### Código fuente de la página:

```

1 <html>
2 <head>
3   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
4 </head>
5 <body>
6   <div id="cajal">
7
8     </div>
9 <h1>titulo 2</h1>
10 <h2>titulo 1</h2>
11
12 <script src="jq02.js" >
13
14
15 </script>
16 </body>
17 </html>

```

### Código del JS externo:

```

$( document.getElementsByTagName("h2")).insertAfter( "#cajal" );

```

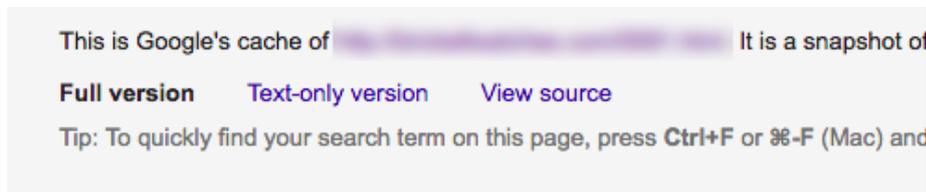
En la versión 'solo texto' del caché de Google, el elemento se encuentra al final de la lista.



**titulo 2**

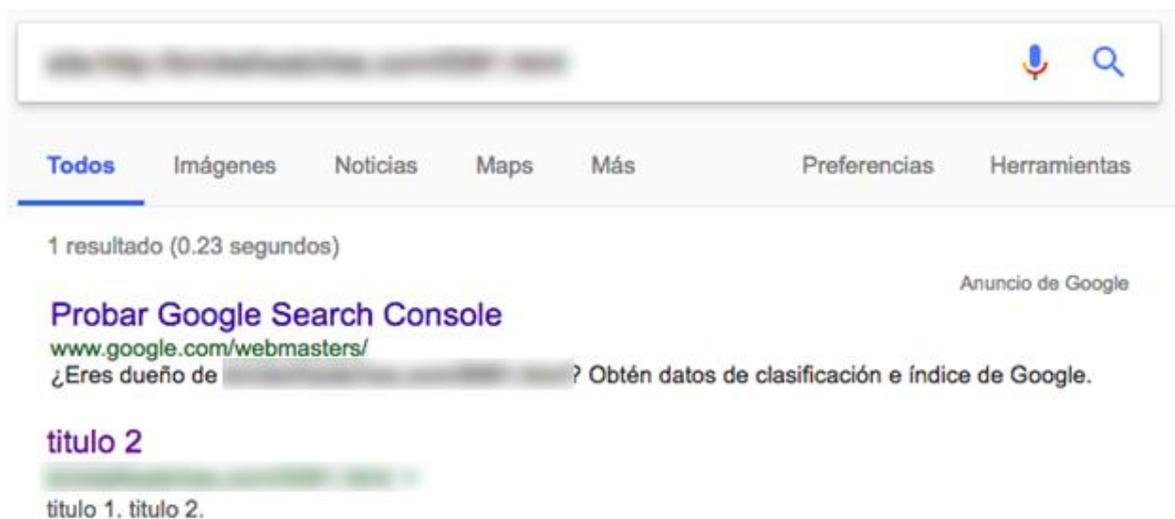
**titulo 1**

Sin embargo, tanto en la versión completa del caché de Google como en el snippet de resultados de búsqueda, el título elemento movido por jQuery se encuentra en primer lugar.



## titulo 1

## titulo 2



### Conclusión:

- Se puede mover contenido para SEO haciendo uso del método "insertAfter" de jQuery localizando el destino por su nombre de etiqueta aunque el código se ejecute desde un archivo .js externo.

## jQuery - Prueba 10

**Tipo de prueba:** Crawleabilidad e indexabilidad de elementos movidos con jQuery

**Objetivo:** Testear si Google interpreta e indexa elementos movidos con jQuery

**Forma de testeo:** Se cambia el orden de un elemento con el método jQuery "prependTo" identificándolo por medio de su tagName. El jQuery se ejecuta desde un archivo .js externo

**Variable:** "Ramiro" esta al final de la lista de nombres. Es identificado por medio de su tag name y movido al primer puesto.

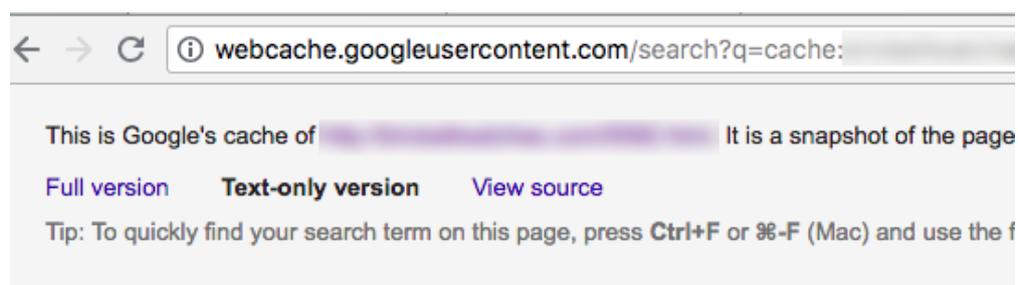
### Código fuente de la página:

```
1 <html>
2 <head>
3     <script src="https://code.jquery.com/jquery-1.10.2.js"></script>
4 </head>
5 <body>
6
7 <ul id="names">
8     <li id="nombre1">Roberto</li>
9     <li id="nombre2">Ruffo</li>
10    <li id="nombre3">Delfina</li>
11    <p>Ramiro</p>
12 </ul>
13
14
15 <script src="jsq03.js">
16 </script>
17 </body>
18
19 </html>
20
```

### Código del JS externo

```
< > ↻ ⓘ 
$( document.getElementsByTagName( "p" ) ).prependTo( "#names" );
```

En la versión 'solo texto' del caché de Google, el elemento se encuentra al final de la lista.



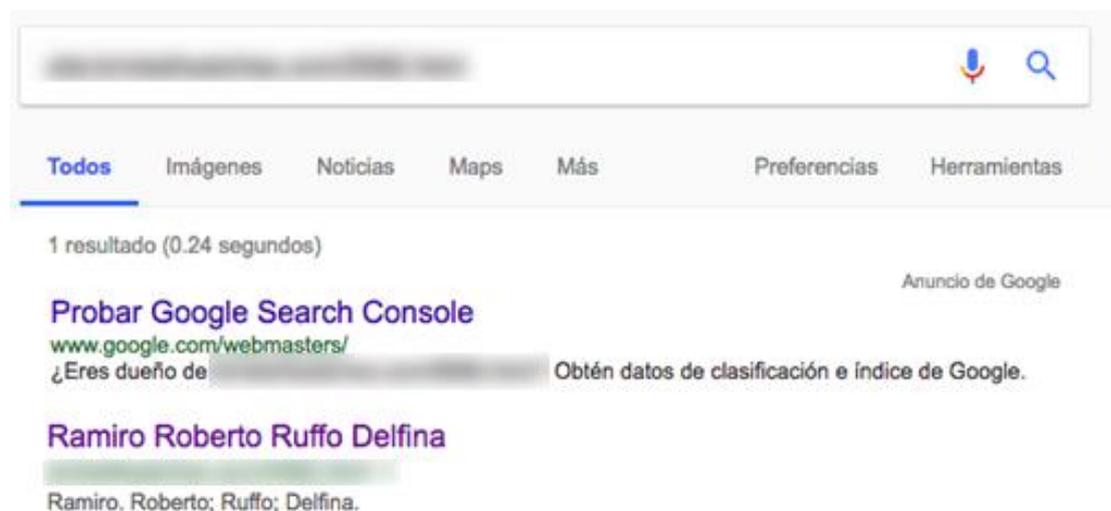
- Roberto
  - Ruffo
  - Delfina
- Ramiro

Sin embargo, tanto en la versión completa del caché de Google como en el snippet de resultados de búsqueda, el título elemento movido por jQuery se encuentra en primer lugar.



Ramiro

- Roberto
- Ruffo
- Delfina



## Conclusiones finales

- Se puede mover contenido para SEO haciendo uso del método `prependTo` de jQuery localizando el destino por su id.
- Se puede mover contenido para SEO haciendo uso del método `insertAfter` de jQuery localizando el destino por su id.
- Se puede mover contenido para SEO haciendo uso del método `insertBefore` de jQuery localizando el destino por su id.
- Se puede mover contenido para SEO haciendo uso del método `insertAfter` de jQuery localizando el destino por su nombre de clase.
- Se puede mover contenido para SEO haciendo uso del método `insertAfter` de jQuery localizando el destino por su id.
- Se puede mover contenido para SEO haciendo uso del método `insertAfter` de jQuery localizando el destino por su nombre de etiqueta.
- Se puede mover contenido para SEO haciendo uso del método `insertBefore` de jQuery localizando el destino por su nombre de etiqueta.
- Se puede mover contenido para SEO haciendo uso del método `prependTo` de jQuery localizando el destino por su nombre de etiqueta.
- Se puede mover contenido para SEO haciendo uso del método `insertAfter` de jQuery localizando el destino por su nombre de etiqueta aunque el código se ejecute desde un archivo `.js` externo.
- Se puede mover contenido para SEO haciendo uso del método `prependTo` de jQuery localizando el destino por su nombre de etiqueta aunque el código se ejecute desde un archivo `.js` externo.

En resumen, tras analizar los casos de prueba, vemos que existen dos tipos de crawling: uno con JavaScript (o, más específicamente, jQuery) y otro sin. El primero es el que se muestra en la página de resultado de búsqueda y el otro en la versión de 'sólo texto'.

**Los textos movidos por JavaScript con métodos jQuery como los testeadas se crawlean e indexan correctamente y se puede contemplar como un recurso para optimizar una página.** Dicho esto, siempre es conveniente que los elementos estén ordenados correctamente en el HTML en lugar de ordenarlos con métodos JavaScript.

Mover elementos vía JavaScript debe utilizarse sólo como un último recurso y en caso de existir obstáculos para ordenar el contenido de forma manual (limitaciones técnicas, de plataforma, falta de tiempo para implementar por los canales tradicionales, etc.).

No dejes de consultar nuestro blog para más [white papers SEO](#).