

Origen y evolución de HTML. El lenguaje SGML y derivados.

Ya en los años 60 del siglo pasado las empresas que se dedicaban a publicar o gestionar documentación electrónica se encontraban con el inconveniente de la falta de compatibilidad: documentos creados con una aplicación no podían ser abiertos o editados con otra aplicación diferente. Esto dificultaba que se pudieran compartir documentos.

Ante esta problemática, la empresa IBM desarrolló un lenguaje de marcas primitivo llamado GML (Generalized Markup Language), que empleaba marcas precedidas del símbolo del doble punto (

para estructurar el documento de forma genérica: listas, tablas y párrafos. Posteriormente, a mediados de los 80, ISO utilizó esta base para definir SGML (Standard Generalized Markup Language), que básicamente es un estándar para definir lenguajes de marcado para documentos. Todos los lenguajes de marcas actuales descienden de él.

Ya en los 90, se define el HTML (HyperText Markup Language) a partir de la sintaxis de SGML. El Consorcio World Wide Web (W3C) regula las recomendaciones y versiones normalizadas de este lenguaje, aceptado como norma ISO desde el año 2000.

HTML fue diseñado originalmente para intercambiar información en entornos académicos, sus etiquetas están pensadas para la organización lógica del contenido (título, párrafo, etc.) y no tanto para su presentación. Por eso, el W3C tuvo que idear las denominadas hojas de estilo (Style Sheets).

En el 2000, aparece el XML (Extensible Markup Language) como complemento al HTML. XML y HTML son lenguajes muy diferentes. Su función principal es ayudarnos a organizar contenidos y eso hace que los documentos XML sean portables hacia diferentes tipos de aplicaciones. Aunque ambos se basan en el SGML, cada uno ha sido diseñado para cumplir distintas funciones: el XML sirve para estructurar información de cualquier tipo mientras que el HTML sirve para estructurar el contenido de texto de un documento web y su metainformación y presentarla a través de un navegador.

Posteriormente surge XHTML (eXtensible HyperText Markup Language), diseñado para extender HTML y permitir la compatibilidad con nuevos formatos de datos como videos, imágenes o lenguajes de scripting. A finales de 2009 se interrumpe el trabajo sobre la versión 2.0 porque el Grupo de Trabajo de Tecnología de Aplicación de Hipertexto Web (WHATWG) estaba trabajando en HTML5 al mismo tiempo, y este último fue el estándar que al final prevaleció. Sin embargo, HTML 5 ya no es una aplicación de SGML, sino un lenguaje generalizado compatible con las versiones anteriores.

DECLARACIÓN DE ETIQUETAS EN LAS DIFERENTES VERSIONES DE HTML

La declaración del tipo de documento en algunas de las distintas versiones de HTML o XHTML quedaría así:

HTML 4.01:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

XHTML 1.1:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

HTML 5:

A partir de HTML 5 la declaración es muy sencilla, ya que no requiere la especificación de una DTD (Document Type Definition en las que se incluyen las reglas sintácticas para un tipo de documento específico) para mostrar la web correctamente, sino más bien, para validarlo.

```
<!DOCTYPE html>
```

Elementos que podemos encontrar en la cabecera

Los elementos que se encuentran en la sección `head` incluyen:

- El título de la página web (`title`)
- Las metaetiquetas (*metatags*) que describen el documento, como la codificación de caracteres utilizada y la información a la que pueden acceder los motores de búsqueda
- Las referencias a archivos externos que necesita la página, tales como scripts y hojas de estilos (`style`, `script`, `base` y `link`). Estos elementos los veremos más adelante en el curso, a medida que los necesitemos.

El elemento `title`

El elemento `title` es el título de la página web; se escribe entre las etiquetas `<title>` y `</title>` y se muestra en la barra de título de la ventana del navegador. Aunque el texto puede ser de cualquier longitud, si sobrepasa los 60 caracteres puede que el navegador lo corte cuando lo muestre.

Las metaetiquetas

Las **metaetiquetas** siempre van dentro del elemento `head`, y se utilizan para indicar el juego de caracteres empleado, la descripción de la página, las palabras clave, el autor del documento y la configuración del área visible de una página (*viewport*), que será diferente para cada tipo de dispositivo.

Las metaetiquetas utilizan dos atributos: el atributo `name`, para indicar el nombre de la metaetiqueta, y el atributo `content`, para indicar su contenido, excepto para la metaetiqueta que indica la codificación de caracteres, que sólo utiliza el atributo `charset`.

```
<head>
  <meta charset="utf-8">
  <meta name="description" content="Ejemplos para 2º SMR">
  <meta name="keywords" content="HTML,WEB">
  <meta name="author" content="Micaela Alfonso">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

Analizando el ejemplo anterior:

Metaetiqueta *charset*

Define la codificación de caracteres para el documento. Una codificación define cómo se traducen los bytes a texto y viceversa. Eligiendo una codificación en particular, decimos cómo debe ser interpretada la secuencia de bytes.

La codificación predeterminada en HTML5 es UTF-8, y cubre (casi) todos los caracteres, signos de puntuación y símbolos del mundo. Si una página web HTML5 utiliza un juego de caracteres diferente al UTF-8, debe especificarse mediante esta meta etiqueta. Por ejemplo, para una web en español se puede utilizar la codificación ISO-8859-1, que define la codificación para Europa occidental:

```
<head>
  <meta charset="iso-8859-1">
</head>
```

Es habitual incluir esta meta etiqueta dentro de los primeros 512 caracteres del documento, para asegurar que todos los navegadores la leen correctamente. Por ejemplo, puede ser la primera etiqueta dentro del bloque `head`, y debemos tener presente que sólo puede haber una (una página HTML sólo puede especificar una única codificación).

Metaetiqueta *description*

Se utiliza para describir brevemente el contenido de una página web. Esta descripción aparece debajo del título y la URL de la página en los resultados del motor de búsqueda. Para permanecer visible dentro de Google, tiene que contener entre 140-160 caracteres. Además, para favorecer un buen posicionamiento es recomendable que describa el contenido de la página de forma significativa y que incluya alguna de sus palabras clave.

Metaetiqueta *keywords*

Define una serie de palabras clave relacionadas con los contenidos de la página web que pueden resultar útiles para los motores de búsqueda, aunque su uso tiene cada vez menor influencia en el posicionamiento. Hoy en día es ignorado por Google y su importancia ha sido minimizada por Bing y otros buscadores populares.

Metaetiqueta *author*

Indica el nombre del autor de la página.

Metaetiqueta *viewport*

Configura el área visible de la página web para se visualice correctamente en todos los dispositivos.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Al añadir el anterior código en una página web le estamos indicando al navegador la siguiente información sobre cómo controlar las dimensiones y la escala de la página:

- `width=device-width` ajusta el ancho de la página al ancho de la pantalla del dispositivo con la que se visualiza la página.

Los elementos que se encuentran en la sección `head` incluyen:

- El título de la página web (`title`)
- Las metaetiquetas (*metatags*) que describen el documento, como la codificación de caracteres utilizada y la información a la que pueden acceder los motores de búsqueda
- Las referencias a archivos externos que necesita la página, tales como scripts y hojas de estilos (`style`, `script`, `base` y `link`). Estos elementos los veremos más adelante en el curso, a medida que los necesitemos.

Un ejemplo sencillo

A continuación se muestra el código HTML de una página web sencilla:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Página de Micaela</title>
    <meta charset="utf-8">
    <meta name="author" content="Micaela Alfonso">
  </head>
  <body>
    <p>Una página sencilla de prueba</p>
  </body>
</html>
```

Comentarios en HTML

HTML permite añadir comentarios en el código. Estos comentarios pueden servir para dos cosas: explicar una parte del código (en este caso, de la estructura o contenidos de la página web) que no quede clara, o para delimitar/marcar una zona del código para poder localizarla o identificarla más fácilmente.

Los comentarios en HTML comienzan con el símbolo `<!--` y terminan con el símbolo `-->`. Lo que haya entre estos símbolos no se muestra en el contenido de la página, sólo se puede leer mirando el código fuente. Veamos un ejemplo:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <!-- Todo lo incluido entre estos caracteres son comentarios y no se ejecuta -->
    <!-- Añadimos el juego de caracteres -->
    <meta charset="utf-8">
    <!-- Título de la página -->
    <title>Página web de Micaela</title>
  </head>
  <body>
    <!-- En este ejemplo, simplemente mostraremos un texto de saludo -->
    <p>Hola, clase</p>
  </body>
</html>
```

Redireccionar el dominio de una web activa

Las redirecciones tienen como función principal notificar a los servidores que el contenido de una página web se ha trasladado de una URL a otra. Esto es fundamental si la dirección original recibe

enlaces entrantes, si ha sido marcada como favorita en el navegador de un visitante o si cuenta con una posición destacada en los resultados de los motores de búsqueda. Así, al redireccionar una web se le está informando al navegador web o al crawler que el contenido buscado ha sido trasladado, redireccionándolo automáticamente a la nueva URL. De no ser así, los visitantes de la página web se encontrarían con el mensaje genérico de error 404, que debe ser evitado, especialmente, por portales de comercio online.

La mayoría de tiendas online ofrecen una gama de productos muy amplia y que cambia regularmente, lo que se refleja en una gran cantidad de páginas web individuales. Así, cuando un artículo no está disponible, los clientes potenciales son simplemente redirigidos a una nueva subpágina con productos similares. Mediante la redirección eficiente de los flujos de visitantes, todo proyecto web está minimizando significativamente la tasa de rebote. Redireccionar una página web permite poner a disposición el mismo contenido web en diferentes URL, redirigiendo todas las direcciones alternativas al dominio deseado.

Aunque nosotros veremos el redireccionamiento con la meta etiqueta de HTML “refresh”, os muestro los tipos que existen para que podáis conocerlos

Tipos de redirección de dominios

En general se distinguen dos tipos de redirecciones: aquellas que se llevan a cabo desde el servidor y aquellas basadas en el cliente. Para redireccionar un dominio desde el servidor, se muestra un código de estado HTTP al agente de usuario (ya sea el navegador web o el crawler). Cuando la redirección se basa en el cliente, se hace de manera automática sin ningún tipo de notificación al visitante. Esto hace que estas últimas no sean compatibles con todos los agentes de usuario, generando así que algunos usuarios permanezcan en la web original sin ser redirigidos al nuevo dominio. Es por este motivo que se suele recomendar redireccionar una página web desde el servidor. Las soluciones basadas en el cliente se deben usar solo cuando las limitaciones técnicas por parte del servidor impiden la redirección.

Redirección desde el servidor

En la mayoría de los casos se utiliza un archivo .htaccess o un script PHP para redireccionar un dominio. La ventaja de estos métodos reside en que es posible definir individualmente qué código de estado HTTP debe mostrarse al agente de usuario, caracterizando así las redirecciones como temporales o permanentes. Así, resultan relevantes especialmente dos códigos de estado HTTP, el 301 y el 302:

- 301 – Moved Permanently: la página solicitada por el agente de usuario estará disponible de manera permanente bajo la URL redireccionada. La antigua URL será, por lo tanto, inválida.
- 302 – Moved Temporarily: la página solicitada por el agente de usuario está temporalmente disponible bajo la nueva URL. A diferencia de la redirección 301, la antigua dirección sigue siendo válida.

En caso de que no se defina específicamente el código de estado HTTP, el servidor mostrará por defecto un código de estado 302, lo que no siempre es ideal. Por lo tanto, es aconsejable introducir manualmente todos los códigos de estado para evitar errores de indexación como el URL Hijacking.

A diferencia de la redirección 301, el código de estado 302 le comunica al crawler que la URL original debe seguir siendo indexada. De mantenerse activa, la URL antigua competirá con la nueva en el índice de los motores de búsqueda.

Consejo SEO

El uso de redirección asegura el traslado adecuado de enlaces. Esto ayuda al sitio web a mantener su ranking y evita una bajada en el tráfico.

Redirección .htaccess

.htaccess hace referencia a un tipo de archivo de configuración para el servidor web Apache, con el que es posible asignar permisos de acceso a directorios y archivos. Estos archivos permiten a los propietarios de páginas web realizar los ajustes necesarios en los directorios de un dominio o un subdominio. Una de las funciones del archivo .htaccess consiste en redireccionar un dominio de una URL a otra desde el servidor. Así, al crear un archivo .htaccess con el siguiente código en el directorio principal, todas las visitas desde la URL antigua serán redireccionadas al dominio “www.ejemplo.es”:

```
Redirect 301 / https://micalsi.es/
```

La línea de código comienza con “Redirect 301” y especifica, por lo tanto, la información que debe transmitirse a partir del código de estado HTTP del servidor. La siguiente parte del código indica que la ruta de acceso guiará al usuario a los contenidos que han sido redireccionados en este caso, como lo denota la barra invertida, a la totalidad de la web. Por último, la línea de comandos incluye la dirección completa de destino a la que el agente de usuario será redirigido.

Este método también permite redirigir archivos individuales. El siguiente código muestra cómo redireccionar un archivo de una página web a otra:

```
Redirect 301 /directorio/documento-de-ejemplo.html http://www.ejemplo.es/ejemplo.html
```

Después del código de estado HTTP 301 se encuentra el directorio que va a ser redirigido de forma permanente (“/directorio/documento-de-ejemplo.html”), así como la URL de destino (“https://micalsi.es”).

En un servidor Apache con el módulo mod_rewrite activado, la redirección permanente de un dominio a otro se verá de la siguiente manera:

RewriteEngine On

```
RewriteRule ^directorio/documento-de-ejemplo.html$ https://micalsi.es [L,R=301]
```

La línea 01 del código activa el módulo mod_rewrite del servidor Apache con el comando “RewriteEngine On”. La segunda línea comienza con “RewriteRule”, que se encarga de definir el enrutamiento del archivo y denotar la dirección de destino. El símbolo de intercalación (^) y el símbolo del dólar marcan el principio y el final de la ruta, L designa la última regla del mod_rewrite para solicitudes similares. R=301 redirige con el código de estado HTTP 301.

Hay que tener especial cuidado con la configuración .htaccess para redireccionar un dominio, pues un error en el código modifica automáticamente los archivos de configuración de la web y puede generar problemas en su funcionamiento. Debido a que las modificaciones son aplicadas

inmediatamente después de haber sido guardadas, es recomendable comprobar el código y verificar los cambios antes de guardarlos.

Redirección con PHP

Redireccionar una página web no solo es posible a través de archivos .htaccess. También lo es usando comandos en script PHP (por ejemplo en el index.php). El siguiente código ejemplifica una redirección permanente a la URL de destino “micalsi.es”.

```
<?php
header("Status: 301 Moved Permanently");
header("Location: https://micalsi.es");
exit;
?>
```

En la redirección con PHP, la función “header” en la línea de código 02 define el código de estado – en este caso el 301 de redirección permanente. Debido a que, por defecto, las redirecciones desde el servidor son temporales, es necesario forzar la redirección a través del código de estado 301. La URL de destino –en este caso “www.ejemplo.es”– en la línea de comando 03 también es definida por la función “header”. La función “exit” en la línea de comando 04 finaliza el script y evita que se ejecuten códigos posteriores. Para que la redirección por PHP funcione, el bloque de código debe ubicarse al principio, asegurando así que el servidor no mostrará ningún tipo de contenido HTML en la página de redirección.

Redirecciones basadas en el cliente

En caso de que, como consecuencia de problemas técnicos, no sea posible redireccionar una página web desde el servidor, alternativamente se puede usar una solución basada en el cliente. Para este fin se dispone de los meta tags HTML “refresh” y del JavaScript correspondiente. La desventaja de la redirección de dominios basada en el cliente es que el servidor no emite ningún código de estado HTTP al navegador web o al crawler. Esto significa que no se hace referencia explícita a la redirección. Por otra parte, no todos los agentes de usuario soportan la redirección basada en el cliente, por lo que se corre el riesgo de que algunos de los visitantes a la página web no sean redirigidos.

Otra de las desventajas es el efecto generado en la indexación de los motores de búsqueda. Al redireccionar una web desde el cliente usando el código de estado HTTP 301, los buscadores no desindexarán automáticamente de sus resultados el dominio antiguo, lo que quiere decir que el anterior dominio empezará a competir en el ranking con el dominio de destino. Mientras que las redirecciones basadas en el cliente siempre implicarán un retraso que seguramente tendrá un impacto en la tasa de rebote de los visitantes, las redirecciones desde el servidor (que se mantienen invisibles para el usuario con la única diferencia del cambio de dirección en la casilla del navegador) resultan mucho más convenientes.

Consejo SEO

Evita la cadenas de redirección, usarlas demasiado puede generar a una bajada de tráfico.

Redireccionar una página web con el meta tag HTML “refresh” (la que usaremos)

La redirección mediante la meta etiqueta HTML se realiza utilizando el atributo “http-equiv”. Así, para configurar la redirección, es necesario crear un archivo HTML simple con la etiqueta correspondiente en el encabezado de la línea de comandos. Con el fin de informar a los visitantes de la web sobre la redirección, es posible crear un mensaje de texto en el documento HTML –p. ej. “¡Nos hemos mudado! Serás redirigido automáticamente en unos segundos”. Una redirección simple de dominios a través del la meta etiqueta HTML “refresh” se verá de la siguiente forma:

```
<meta http-equiv="refresh" content="10; url=http://www.ejemplo.es/">
```

La meta etiqueta “http-equiv” se encarga de solicitarle al cliente la redirección del dominio. A continuación se define el atributo del contenido. En el comando anterior, pasados 10 segundos, el usuario será redirigido a la URL de destino “www.ejemplo.es”.

Redirección con JavaScript

Utilizar JavaScript es otro método sencillo para redireccionar una URL desde el cliente. Al igual que la redirección con la etiqueta meta “refresh”, la redirección con JavaScript solo debe ser usada como último recurso, pues por motivos de seguridad este script no es soportado por todos los navegadores web. Recuerda además que las redirecciones con JavaScript también son un problema para crawlers y usuarios que tengan activados complementos NoScript en sus navegadores. Un ejemplo clásico de redirección a través de JavaScript se verá de la siguiente forma:

```
<script>
<!--
window.location.replace('https://micalsi.es');
//-->
</script>
```

La importancia en esta secuencia de comandos reside principalmente en la línea 03. En el ejemplo, el objeto “window.location” se utiliza para referirse al nuevo dominio. El comando “replace” le indica al navegador que debe redirigir al usuario al dominio entre paréntesis “micalsi.es”.